Using Auxiliary Sensors for Pairwise Key Establishment in WSN

Qi Dong and Donggang Liu

Department of Computer Science and Engineering
The University of Texas at Arlington
Box 19015, Arlington, Texas 76019-0015, USA
{qi.dong,dliu}@uta.edu

Abstract. Many techniques have been developed recently for establishing pairwise keys in sensor networks. However, they are either vulnerable to a few number of compromised sensor nodes or involve expensive protocols for establishing keys. This paper introduces a much better alternative for achieving high resilience to node compromises and high efficiency in key establishment. The main idea is to deploy additional sensor nodes, called *assisting nodes*, to help the key establishment between sensor nodes. The proposed approach has many advantages over existing approaches. In this approach, a sensor node only needs to make a few local contacts and perform a few hash operations to setup a key with any other sensor node in the network at a very high probability. The majority of sensor nodes only need to store a single key in their memory space. Besides these benefits, it still provides high resilience to node compromises. The implementation of this approach on TelosB motes also demonstrates its feasibility for pairwise key establishment in sensor networks.

Key words: key management, pairwise keys, sensor networks

1 Introduction

Wireless sensor networks are ideal candidates for a wide range of applications in military and civilian operations such as health monitoring, data acquisition in hazardous environments, and target tracking. Security has been recognized as a critical requirement for many sensor applications, especially in military operations. Key management is the cornerstone to ensure the security of many network operations. As one of the most fundamental security services, pairwise key establishment enables secure node-to-node communication using cryptographic methods such as encryption and authentication.

Many techniques have been developed recently to setup pairwise keys in sensor networks [1–10]. Perrig et al. developed the SNEP protocol to provide pairwise key establishment using a KDC [1]. This approach, however, introduces huge communication overhead and is vulnerable to single point failure. A number of key pre-distribution schemes were proposed to establish keys without the

online KDC [2–6]. These approaches preload a small set of secrets into every sensor node before deployment to make sure that after deployment, every two sensor nodes can setup a shared key using their preloaded secrets. However, these approaches either require expensive protocols (e.g., path key establishment) to setup keys or are vulnerable to a small number of compromised sensor nodes. In addition, some techniques also use the sensors' location information and assume static sensor nodes [7, 8, 11, 10, 9]. However, these two assumptions may not be true in practice.

This paper presents a novel technique for pairwise key establishment in sensor networks. The main idea is to deploy additional sensor nodes, called *assisting nodes*, to help the key establishment between sensor nodes. Different from the nodes in traditional networks where they are mainly used for sensing and forwarding, the assisting nodes are only responsible for key management in the network, exploiting a novel dimension of using sensor nodes. The proposed approach has many advantages over existing approaches. First, it can achieve a very high probability of establishing a shared key between any two sensor nodes. Second, a sensor node only needs to make a few local contacts and perform a few hash operations to setup a key with any other sensor node in the network. Third, the majority of sensor nodes only need to store a single key in their memory space. Fourth, it does not depend on the sensors' location information and can be used for the sensor networks with highly mobile sensor nodes. Besides these benefits, our approach still provides high resilience to node compromises. Finally, the implementation of this approach on TelosB motes [12] also demonstrates its feasibility for key establishment in sensor networks.

The rest of the paper is organized as follows. Section 2 presents our pairwise key establishment protocol as well as the detailed analysis. Section 3 gives the implementation issues. Section 4 reviews related work. Section 5 concludes this paper and points out some future work.

2 Pairwise Key Establishment

This section provides the technical detail as well as the analysis on how to establish pairwise keys using auxiliary sensors. In this paper, we consider the sensor networks consisting of a large number of tiny resource-constrained sensor nodes [13]. These sensor nodes can be static or highly mobile. We assume that the attacker can eavesdrop, modify, forge, replay or block any network traffic. We also assume that the attacker can compromise a few sensor nodes and learn all the secret information, including the keying materials, on those compromised nodes [14].

2.1 Protocol Description

Typically, sensor nodes are deployed to sense the conditions in their local surroundings and report observations for various uses. However, in this paper, we exploit a new dimension of using sensor nodes and believe that it is important

to deploy sensor nodes to facilitate certain network protocols such as key management. Hence, the main idea of our approach is to deploy additional sensor nodes, called *assisting nodes*, to help the pairwise key establishment between sensor nodes. The detailed protocol is presented below. Let n be the network size and m be the number of assisting sensor nodes. For convenience, we call the sensor nodes that are not assisting nodes as the *regular sensor nodes*.

- *Initialization:* Before deployment, the base station generates a master key K_u for every sensor node u. The master key K_u is only known by the sensor node u and the base station. Every assisting node i will get preloaded with a hash $H(K_u||i)$ for every regular sensor node u, where H is a one-way hash function, and "||" denotes the concatenation operation. Hence, an assisting node will need to store n hash images. This clearly introduces considerable storage overhead at assisting sensor nodes. However, the only job of the assisting nodes is to help pairwise key establishment. As a result, they can use all their memory, including the flash memory, to store these values. Therefore, we believe that it will be feasible for an assisting node to store n hash images. For instance, the TelosB motes have 1MB flash memory and can store the hash images for a network of 128,000 sensor nodes if every hash is 8 bytes long. Additionally, research focusing on high-capacity and energy-efficient storage subsystem on sensor network platforms has drawn a lot of attention, which will soon make it possible to equip a sensor node with a large flash memory [15] without increasing the cost significantly. Therefore, more and longer hash images can be stored in each assisting node for a very large sensor network.
- *Pairwise Key Establishment:* After deployment, every regular sensor node discovers the assisting nodes in its neighborhood. When a sensor node u needs to establish a pairwise key with another node v, it will send a request to every neighbor assisting node i. The request message includes the IDs of both sensor nodes and will be protected by the key $H(K_u||i)$, which has been preloaded to the assisting node i. The assisting node i will serve as a KDC and generate a reply to u. This reply message includes two copies of a random key R, one is protected by $H(K_u||i)$ (for node u) and the other is protected by $H(K_v||i)$ (for node v). This procedure is similar to the Needham-Schroeder Symmetric Key Protocol[16]. After the request, u will get a random key from every neighbor assisting node. Let $\{R_1, ..., R_l\}$ be the set of all these random keys. The final key $K_{u,v}$ between u and v is simply the bit-wise XOR of all these keys, i.e., $K_{u,v} = R_1 \oplus R_2 \oplus \cdots \oplus R_l$. Obviously, as long as at least one random key is secure, the final key will be safe.

Though our later analysis in Section 2.2 shows that even a small fraction of assisting nodes can guarantee a high probability of establishing pairwise keys using the above algorithm, it is still possible that a regular sensor node cannot find any assisting sensor node in its neighborhood since the accurate deployment of assisting nodes may not be guaranteed in some scenarios. To deal with this issue, we have *supplemental key establishment*, where a regular sensor node may

discover the set of assisting nodes within a certain number of hops. This will certainly increase the chance of finding an assisting node to use. An additional benefit of doing this is to achieve better security performance. From the previous description, u derives the final key by applying XOR operations to all the random keys, which implies that the more random keys, the higher the security of the final key.

- *Supplemental Key Establishment:* In this step, a sensor node u discovers the assisting sensor nodes that are no more than h hops away from itself. This can be easily achieved by having node u's neighbors to help collecting the IDs of the assisting nodes around them. The neighbor nodes will broadcast the inquiry message on behalf of u, and forward u the replies from assisting nodes. Once such set is discovered, the remaining step will be similar to the pairwise key establishment discussed before.
  The discovery and usage of assisting nodes multiple hops away will introduce additional communication overhead since the intermediate nodes will be needed to relay the messages. However, this will only involve communication in a local area, which we believe will not be a big problem for the current generation of sensor networks. Also, the need for the supplemental key establishment will not likely be invoked frequently.

2.2 Analysis and Discussion

This subsection will present the performance analysis of the proposed scheme, focusing on the probability of establishing pairwise key, the resilience against node captures, and the overheads. For simplicity, we assume that all the n regular sensor nodes and the m assisting nodes are evenly deployed in the field.

Probability of Establishing Keys: During the pairwise key establishment, a sensor node u is required to communicate with at least one assisting node in its neighborhood to setup a key with another sensor node. Let d denote the average number of one-hop neighbors of a sensor node. The probability that any assisting node i is not in the local area of the regular sensor node u can be estimated by $1 - d/(m+n)$. Thus, the probability that a regular sensor node fails to find any assisting node in its neighborhood can be estimated by $(1 - d/(n+m))^m$. The probability P of establishing a pairwise key can then be estimated by $P = 1 - (1 - d/(n+m))^m$. As $d << n$ is usually satisfied in practice, P can be further approximated by $1 - e^{-dm/(n+m)}$. Figure 1 shows the relationship between the probability P of establishing key and the fraction of assisting nodes $\frac{m}{n}$. From the figure, we can see that a small fraction (0.1) can guarantee a high probability (greater than 0.9) of establishing keys between sensor nodes. On the other hand, since P increases faster with larger number of d , the proposed scheme can achieve attractive performance in high-density networks.

During the supplemental key establishment, the assisting nodes within h-hop range of u will be used. Obviously, the larger the range, the higher probability of finding the assisting nodes. Here we only analyze the situation when $h = 2$. From

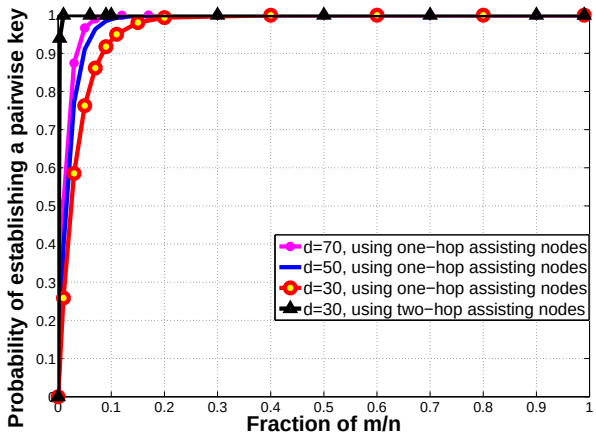

Fig. 1. The probability P that two sensor nodes can establish a pairwise key v.s. the fraction of assisting sensor nodes $\frac{m}{n}$.

the above discussion, the probability that none of u's neighbor nodes can find any assisting node can be estimated by $((1-d/(n+m))^m)^d$. Let P' denote the probability of establishing a pairwise key using the assisting nodes in two-hop range. Therefore, $P' = P + (1-P) \times (1-(1-d/(n+m))^{md})$. Figure 1 indicates that by using two-hop assisting nodes, even when the fraction of $\frac{m}{n}$ is as small as 0.003, the probability of establishing key is still greater than 0.9.

Resilience against Node Captures: We assume that the base station will never be compromised. Thus, the master key of any non-compromised node will be always safe since the assisting nodes are only equipped with the hashes of the master key. Even if an assisting node is captured, it is computationally infeasible to get the original keys from the hashes due to the one-way property of hash functions.

We also note that the sensor node derives the final key by applying XOR operation to all the random keys. The pairwise key will be secure unless all the related assisting nodes are compromised. This indicates an attractive property of our scheme: *a benign assisting node can guarantee the security of the keys established in its neighborhood as long as it can communicate with the sensor nodes in its neighborhood.*

We then study the probability P_c of a key being compromised when a certain fraction of nodes are compromised. Assume the attacker will randomly compromise a fraction f_c of sensor nodes. From our earlier analysis, P_c is equal to the probability that all the corresponding random keys are compromised. The number of assisting nodes that might provide those random keys can be estimated by $\frac{m \times d}{n}$. Hence, P_c can be estimated by $f_c^{(m \times d)/n}$. Figure 2 shows that our approach is highly resilient to the node compromise attack. It also implies that we can enhance the security of key management by deploying more assisting nodes.

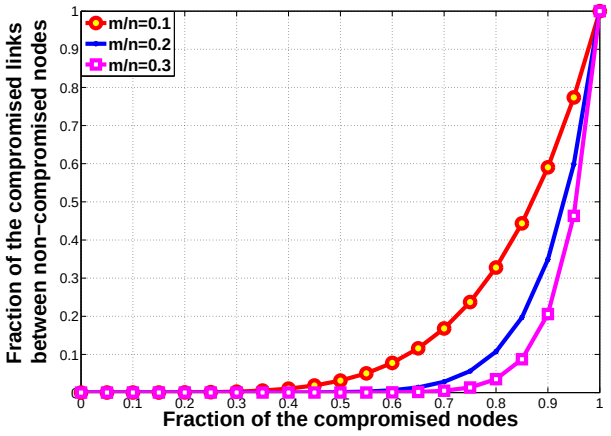

Fig. 2. The fraction of compromised links between non-compromised nodes v.s. the fraction of compromised nodes.

Overheads: The proposed scheme only requires a single key for every regular sensor node and n hash values for an assisting node. As we discussed before, the assisting node is only responsible for the pairwise key establishment and can use all its memory, including the flash memory, to store these hash images.

The proposed scheme involves small computation overheads. A regular sensor node only need to apply a few symmetric key operations and hash operations to establish a key with any other sensor node.

We note that in pairwise key establishment phase, the communication is in one-hop range. Although multi-hop communication will occur when supplemental key establishment is needed, the communication is still limited in a local area. Moreover, we have shown that in most cases, only one-hop communications are needed. As a result, the supplemental key establishment will not incur significant communication overhead for our protocol.

2.3 Comparison with Previous Schemes

This section will compare the proposed scheme with previous techniques for pairwise key establishment such as the basic probabilistic scheme [2], the q-composite scheme [3], the random pairwise keys scheme [3], the random subset assignment scheme [4], and the grid-based scheme [4].

Security Performance: We assume the network size $n = 20,000$ and the average number of neighbors $d = 50$. For previous schemes, we assume each sensor can store 200 keys. Hence, for the grid-based scheme [4], the probability of two nodes sharing a direct key is 0.014. For the other schemes, we set $P = 0.33$ to make sure the network is well connected. However, from our previous analysis, our approach can always guarantee the establishment of pairwise keys at a very high probability using a small fraction of assisting nodes.

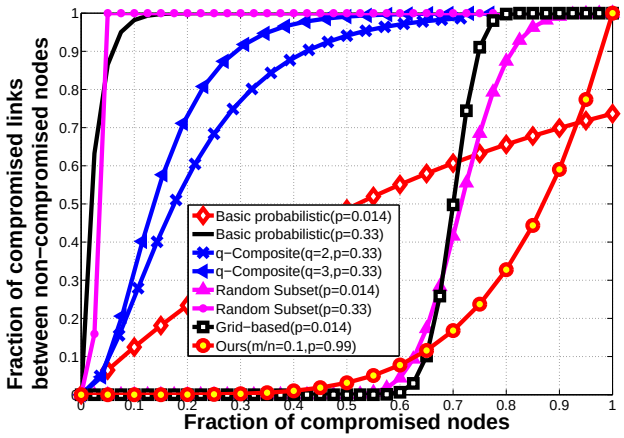

Fig. 3. The fraction of compromised links between non-compromised nodes in different schemes.

Figure 3 shows the fraction of compromised links in the presence of compromised nodes for different schemes. The figure tells us that in terms of protecting the direct keys, our scheme can provide high resilience to node compromises, which is similar to the random subset assignment scheme and the grid-based scheme [4]. In addition, we must remember that our scheme can guarantee a much higher probability of establishing a pairwise key between two sensor nodes in a densely deployed sensor network.

Note that the previous schemes need to employ expensive protocols for path key establishment when two sensor nodes cannot directly setup a pairwise key. As a result, the attacker might discover not only the direct keys but also the indirect (path) keys by compromising the intermediate nodes used in the establishment of the indirect (path) keys. On the contrary, our approach does not need to setup path keys. Even if the attacker has captured the nodes which relay the keying information, the key will not be disclosed. Figure 4 shows the fraction of compromised (direct or indirect) keys between non-compromised nodes in the presence of compromised nodes. The figure clearly shows that our scheme performs much better than other schemes. For example, when 70% sensor nodes are compromised, the fraction of compromised pairwise keys between non-compromised sensor nodes is only around 0.18. Contrarily, at least 88% keys have been exposed in the previous schemes.

Additionally, our proposed scheme can guarantee that a single benign assisting node can protect the keys established in its neighborhood as long as this node can talk to the sensor nodes in its neighborhood.

Overheads: In the proposed scheme, only a single master key is stored in every regular sensor node, while the previous schemes have considerable storage requirements for achieving high performance. For instance, In Figure 3 and Fig-

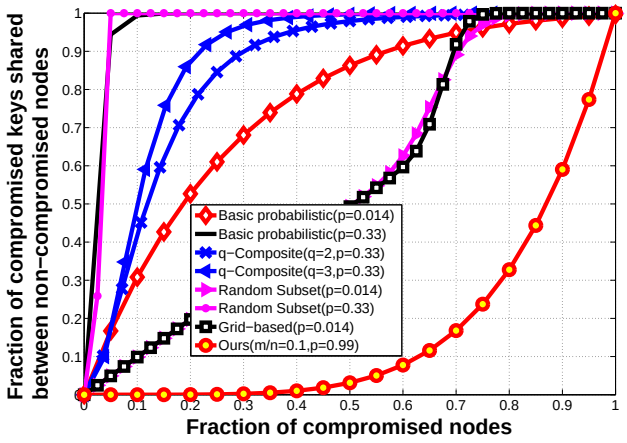

Fig. 4. The fraction of compromised (direct or indirect) keys between non-compromised nodes in different schemes.

ure 4, the previous schemes require every node to store 200 entries to achieve the desire performance. In terms of the computation overhead, our proposed approach involves only a few number of symmetric key operations and hash operations. Hence, it will not incur much additional overhead.

From the previous discussion, most of the communication overhead is introduced in the pairwise key establishment phase, which only requires direct communications in one-hop range. The multi-hop communication in the supplemental key establishment rarely occurs due to the high probability of finding a neighbor assisting node. On the other hand, for many previous schemes such as the grid-based scheme [4] and PIKE [6], a sensor node often needs to go through the path key establishment to setup keys with other sensor nodes. Such path key establishment could be very expensive in practice since the intermediate sensor node that can help establish the pairwise key may be located far away from the two sensor nodes that want to establish a shared key. According to the above discussion, we can clearly see that our proposed approach has significant advantages over existing schemes in terms of storage, computation and communication overheads.

2.4 Security Reinforcing Version

From our previous analysis, we note that once an assisting node is compromised, the attacker is able to discover all the random keys generated by this node. Although the actual pairwise key is combined from multiple random keys generated by different assisting nodes, it is still not desirable to let the attacker figure out the old random keys. In the following, we give a simple extension to fix this problem by updating the keys at the assisting nodes.

The basic idea is to update the key at every assisting sensor node after the pairwise key establishment. In other words, the hash key at any assisting node will be changed immediately after it is used once. As a result, the attacker is not able to reveal any random key generated before even if the assisting sensor node is compromised at certain point. To achieve this goal, we will take the advantage of the one-way hash function $H(\cdot)$. We will also maintain a sequence number for every hash key shared between a regular sensor node and an assisting node. For example, initially, the hash key $H_u = H(K_u||i)$ and the sequence number $S_u = 0$ will be stored in the assisting node i for the regular sensor node u before deployment.

Once the assisting node i receives the request message to setup a pairwise key between u and v, i will send a random key R to one of the nodes along with the current sequence number S_u and S_v via the secure links. At this time, the two copies of the random key R are protected by the hash values H_u and H_v respectively. After that, node i will replace H_u with $H(H_u)$ and H_v with $H(H_v)$, and also increase S_u and S_v by 1.

When the regular sensor node u receives the message from i, u will verify the authentication and confidentiality of the message using the same hash H_u, which can be computed based on K_u, i and S_u. Node u can then derive the random key R for pairwise key establishment. In the protocol, node u may choose to keep track of the sequence number S_u with each neighbor assisting node in its local area to reduce the overhead during the calculation of the hash key H_u. Such sequence number can certainly be used to deal with the replay attacks as well.

Therefore, by employing one-way hash function, the improved approach can enhance the resilience against node captures, i.e., any compromised node will not reveal any secret about the pairwise keys established before. However, compromised assisting sensor nodes can still participate in the pairwise key establishment in the future when new nodes are added in the network. These malicious assisting nodes may disclose valuable information to adversaries. Fortunately, our scheme guarantees that as long as there are at least one benign assisting node in a given area, the final pairwise key will be still safe no matter how many sensor nodes are compromised. Based on this property, we may deploy new assisting sensor nodes to replace the old and untrustworthy ones to achieve better security during the pairwise key establishment.

3 Implementation Issues

Based on the previous analysis, we can see that our proposed pairwise key establishment approach is efficient for resource-constrained sensor nodes. In this section, we will present the implementation issues.

We have implemented the prototype of the proposed scheme under TinyOS platform [17]. We use RC5 module [18] to implement the security primitives such as hash and MAC operations, assuming 8-byte long hash values and keys. This mechanism is transparent to the applications. In the protocol, the regular sensor node will first send request messages to its neighbor assisting nodes and wait for

their responses. An assisting node generates the random key and send the reply message to every requesting sensor node. After the regular sensor node collects the random keys, it will combine these keys to derive the final pairwise key.

Our scheme has been tested on the TelosB [12] motes. For the assisting nodes, the additional code space is 2598 bytes in the ROM, and the extra usage of data space in the RAM is 892 bytes. We also make use of the 1M flash memory on the chip to store the hash values of the regular nodes' master keys. For the regular nodes that need to setup secure communication links with 50 neighbors, the additional code space in the ROM is 2102 bytes, and the extra data space in the RAM is 682 bytes. Clearly, the proposed scheme is practical for sensor networks in terms of the code size.

4 Related Work

The closest related work to the techniques studied in this paper is pairwise key establishment. Many techniques have been proposed along this research direction, including the basic probabilistic scheme [2], the q-composite scheme [3], the random pairwise keys scheme [3], the two threshold-based schemes [4, 5] and PIKE [6]. Additionally, the prior deployment and post deployment knowledge were also used to improve the performance of key pre-distribution in various situations [7, 8, 11, 10]. This paper gives a better way for pairwise key establishment by exploiting a new dimension of using sensor nodes.

There are many other studies on sensor network security, including broadcast authentication [1, 19], tamper-resistant hardware [20], secure data aggregation [21], and vulnerabilities, attacks, and countermeasures [22]. We consider them complementary to our technique.

5 Conclusion and Future Work

In this paper, we developed a novel scheme to establish the pairwise keys in sensor networks. This scheme takes advantage of special nodes (the assisting nodes) in the network for key management, representing a new dimension of using sensor nodes. The analysis indicates that our scheme has significant advantages over the existing approaches. By making use of these cheap assisting nodes, we ease the burdens of other regular sensor nodes and further extend the lifetime of the whole network.

Several research directions are worth investigating. We are interested in conducting a thorough evaluation in a large scale sensor network. We are particularly interested in issues on how to tolerate the communication error and delay in bad channel conditions, how to withstand the high deployment failure rate where a lot of multi-hop communication may be needed. In addition, the additional assisting nodes in the proposed scheme are only deployed to establish the pairwise keys. However, we may make further use of those nodes to defend the network against various other attacks.

Acknowledgment The authors would like to thank the anonymous reviewers for their valuable comments.

References

1. Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, D.: SPINS: Security protocols for sensor networks. In: Proceedings of Seventh Annual International Conference on Mobile Computing and Networks. (July 2001)
2. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: Proceedings of the 9th ACM Conference on Computer and Communications Security. (November 2002) 41–47
3. Chan, H., Perrig, A., Song, D.: Random key predistribution schemes for sensor networks. In: IEEE Symposium on Research in Security and Privacy. (2003) 197–213
4. Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03). (October 2003) 52–61
5. Du, W., Deng, J., Han, Y.S., Varshney, P.: A pairwise key pre-distribution scheme for wireless sensor networks. In: Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03). (October 2003) 42–51
6. Chan, H., Perrig, A.: PIKE: Peer intermediaries for key establishment in sensor networks. In: Proceedings of IEEE Infocom. (March 2005)
7. Du, W., Deng, J., Han, Y.S., Chen, S., Varshney, P.: A key management scheme for wireless sensor networks using deployment knowledge. In: Proceedings of IEEE INFOCOM'04. (March 2004)
8. Liu, D., Ning, P.: Location-based pairwise key establishments for static sensor networks. In: 2003 ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN '03). (October 2003) 72–82
9. Liu, D., Ning, P.: Improving key pre-distribution with deployment knowledge in static sensor networks. ACM Transaction on Sensor Networks (TOSN) **1**(2) (2005)
10. Yu, Z., Guan, Y.: A key pre-distribution scheme using deployment knowledge for wireless sensor networks. In: Proceedings of ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). (April 2005)
11. Huang, D., Mehta, M., Medhi, D., Harn, L.: Location-aware key management scheme for wireless sensor networks. In: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks (SASN '04). (October 2004) 29 – 42
12. Crossbow Technology Inc.: Wireless sensor networks. `http://www.xbow.com/Products/Wireless_Sensor_Networks.htm` Accessed in February 2006.
13. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: A survey. Computer Networks **38**(4) (2002) 393–422
14. Hartung, C., Balasalle, J., Han, R.: Node compromise in sensor networks: The need for secure systems. Technical Report CU-CS-990-05, U. Colorado at Boulder (Jan. 2005)
15. Mathur, G., Desnoyers, P., Ganesan, D., Shenoy, P.: Ultra-low power data storage for sensor networks. In: Information Processing in Sensor Networks, 2006(IPSN 2006). (April 2006)
16. Needham, R.M., Schroeder, M.D.: Using encryption for authentication in large networks of computers. Commun. ACM **21**(12) (1978) 993–999

17. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K.S.J.: System architecture directions for networked sensors. In: Architectural Support for Programming Languages and Operating Systems. (2000) 93–104
18. Rivest, R.: The RC5 encryption algorithm. In: Proceedings of the 1st International Workshop on Fast Software Encryption. Volume 809. (1994) 86–96
19. Liu, D., Ning, P.: Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks. In: Proceedings of the 10th Annual Network and Distributed System Security Symposium (NDSS'03). (February 2003) 263–276
20. Basagni, S., Herrin, K., Bruschi, D., Rosti, E.: Secure pebblenets. In: Proceedings of ACM International Symposium on Mobile ad hoc networking and computing. (2001) 156–163
21. Przydatek, B., Song, D., Perrig, A.: SIA: Secure information aggregation in sensor networks. In: Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys '03). (Nov 2003)
22. Wood, A.D., Stankovic, J.A.: Denial of service in sensor networks. IEEE Computer **35**(10) (2002) 54–62