KEY ESTABLISHMENT IN SENSOR NETWORKS BASED ON TRIANGLE GRID DEPLOYMENT MODEL

Yun Zhou, Yanchao Zhang, and Yuguang Fang
Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611
Tel: (352)392-8576; Fax: (352)392-0044
Email: {yzufl@, yczhang@, fang@ece.}ufl.edu

ABSTRACT

Key establishment between neighboring nodes is a fundamental issue for securing sensor networks deployed in unattended and hostile environments. Recent research often assumed a probabilistic approach by pre-distributing a random set of secrets to each node such that any two neighboring nodes can establish a direct key with a certain probability or negotiate an indirect key through multi-hop routing after deployment. The shortcomings include poor resilience against node compromise, low secure connectivity, etc. In this paper, we present a key establishment scheme based on a triangle grid deployment model. In contrast to previous proposals, our scheme have higher resilience to node compromise attacks with much smaller memory costs, while maintaining high secure connectivity so that much less energy needs to be consumed in establishing indirect keys through multi-hop or multi-path routing.

INTRODUCTION

A wireless sensor network usually consists of hundreds to thousands of resource-limited sensor nodes deployed in a designated area without any fixed infrastructure [1]–[3]. It is vulnerable to malicious attacks in unattended and hostile environments such as battlefield surveillance and homeland security monitoring [4], [5]. For instance, adversaries can easily eavesdrop messages transmitted over the air between nodes, or disable the entire network by launching physical attacks to sensor nodes or logical attacks to communication protocols [6], [7]. There are a lot of proposals [8]–[14] try to provide encryption and authentication services in sensor networks. Due to the infeasibility of applying public key techniques in infrastructureless sensor networks [8], all the proposals are based on symmetric key techniques.

Eschenauer and Gligor [9] (E-G hereafter)first proposed to use probabilistic key pre-distribution to establish pairwise keys between neighboring nodes. In their scheme, each node is preloaded with a key subset from a global key pool in such a way that any two neighboring nodes can share at least one common key with a certain probability. This scheme is vulnerable to the *node compromise attack*, where keys held by normal nodes can be exposed when some nodes are compromised by adversaries. Chan, Perrig, and Song proposed the *q-composite random key pre-distribution* scheme [10], in which they modified E-G scheme by only increasing the number of keys that two random nodes share from at least 1 to at least q. Their scheme achieves greatly strengthened security under small scale attacks while trading off increased vulnerability in the face of a large scale node compromise attack.

Du *et al.* proposed the *multiple-space key pre-distribution* scheme (MSK hereafter), where each key in [9], [10] is replaced by a special key space. After deployment, any pair of neighboring nodes can establish a pairwise key if they have a common key space. Their scheme supports both encryption and authentication in each key space and has better resilience to the node compromise attack compared with [9], [10].

Chan and Perrig proposed a pairwise key establishment scheme called *PIKE* [12], in which each node has unique pairwise keys with $2(\sqrt{n}-1)$ peer nodes, where n is the total number of nodes in the network. Each node can establish pairwise keys with its neighboring nodes with the help of one peer node.

All previous schemes assume a random node deployment model. As a result, each node can just have direct pairwise keys shared with a portion of its neighbors, and have to rely on multi-hop or multi-path routing to establish indirect pairwise keys with the other neighbors, thus leading to unfavorable low local secure connectivity, which is the probability that any two neighboring nodes share one direct key.

Du *et al.* [13] proposed to utilize node deployment knowledge to improve the local secure connectivity. Their scheme assumes a group-based deployment model, in which the entire network is divided into many non-overlapping square cells and in each cell a group of sensor nodes is deployed. The Eschenauer-Gligor scheme is applied in each cell.

In this paper, we propose a novel key establishment scheme by combining Blom's scheme [15] and node deployment knowledge. We use a triangle grid to model node deployment, thus dividing the entire network into many non-overlapping triangle cells, and

This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N000140210464 and under grant N000140210554.

apply Blom's scheme in each pair of neighboring cells. Compared with conventional proposals, our scheme has perfect resilience to the node compromise attack with low memory cost due to the t-*secure* property of Blom's scheme. Local secure connectivity is high because of the usage of node deployment knowledge.

The rest of the paper is organized as follows. Section II describes the Blom's scheme. Section III gives the details of our scheme. Some analysis and performance evaluation are carried out in Section IV. The paper is concluded in Section V.

BLOM'S SCHEME

Blom [15] proposed a key distribution method that allows any pair of users in a system to be able to find a unique shared key. The optimality of his method was based on $(N, t+1)$ MDS linear codes [16] in that in a system with N users the collusion of less than $t+1$ users can not reveal any key held by other normal users, i.e., t-*secure*. The memory cost per user of Blom scheme is $t+1$. To guarantee perfect secure in a system with N users, the $(N-2)$-secure Blom scheme should be used, which means the memory cost per user is $N-1$.

During the initialization phase, a central authority first constructs a $(t+1) \times N$ matrix P over a finite field $GF(q)$, where N is the size of the network. P is known to all users. Then the central authority selects a random $(t+1) \times (t+1)$ symmetric matrix S over $GF(q)$, where S is secret and only known by the central authority. An $N \times (t+1)$ matrix $A = (S \cdot P)^T$ is computed, where $(\cdot)^T$ is the transpose operator. Because S is symmetric, it is easy to see:

$$\begin{aligned} K = A \cdot P &= (S \cdot P)^T \cdot P = P^T \cdot S^T \cdot P \\ &= P^T \cdot S \cdot P = (A \cdot P)^T = K^T \end{aligned} \quad (1)$$

User pair(i, j) will use K_{ij}, the element in row i and column j in K, as the shared key. Because K_{ij} is calculated by the i-th row of A and the j-th column of P, the central authority assign the i-th row of A and the i-th column of P to each user i, for $i = 1, 2, \ldots, N$. Therefore, when user i and user j need to establish a shared key between them, they first exchange their columns of P, and then they can compute K_{ij} and K_{ji}, respectively, using their private rows of A.

It has been shown in [15] that as long as any $t+1$ columns of P are linearly independent the desirable t-secure property can be achieved.

DETAILS OF OUR SCHEME

A. Adversary Model

Due to the broadcast characteristic of wireless radio, adversaries can easily eavesdrop any messages, non-encrypted or encrypted, transmitted over the air between nodes. Besides, due to cost constraints, it is also unrealistic and uneconomical to employ tamper-resistant hardware to secure the cryptographic materials in each individual node. Hence adversaries may capture any node and compromise the secrets stored in the node. Furthermore, adversaries can use the compromised secrets to derive more secrets stored in other normal nodes. It means that the node compromise attack is unavoidable. What we can do is to reduce the impact on other normal nodes as much as possible when some nodes are compromised.

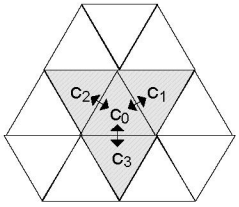

Figure 1. A triangle grid deployment model.

B. Deployment Model

In our scheme, the entire network is divided into many non-overlapping triangle cells (Fig. 1). In each cell a group of N_c nodes is deployed. Usually nodes in each cell reside according to some probability distribution function (PDF). Here we assume in each cell nodes are uniformly deployed.

C. Pre-deployment Phase

In the pre-deployment phase, each node is preloaded with some secrets and those secrets are used to establish shared keys during the post-deployment phase.

1) Construct Public Matrix: Suppose there are N sensor nodes to be deployed and there N_c nodes in one cell. Each node has a positive identifier n_i for $i = 1, 2, \ldots, N$. A $(t+1) \times N$ public matrix P can be constructed as

$$P = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ n_1 & n_2 & n_3 & \cdots & n_N \\ {n_1}^2 & {n_2}^2 & {n_3}^2 & \cdots & {n_N}^2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ {n_1}^t & {n_2}^t & {n_3}^t & \cdots & {n_N}^t \end{bmatrix} \quad (2)$$

Here we choose $t = 2N_c - 2$. The reason of this choice is to be stated later. Since P is a *Vandermonde* matrix, it can be shown that any $t+1$ columns of P are linearly independent when $n_i, i = 1, 2, \ldots, N$ are all distinct [16].

2) Construct Secret Matrices: For each pair of neighboring cells in the network, a secret $(t+1) \times (t+1)$ symmetric matrix S_i over $GF(q)$ is constructed, where $i = 1, 2, \ldots, N_S$ and N_S is the total number of pairs of neighboring cells. S_i is only known by the central authority. S_i is used to derive secrets that are preloaded into the nodes in the corresponding pair of neighboring cells. For example, in Fig. 1, the pair of cells (C_0, C_1) is assigned a secret matrix, so are the pair of (C_0, C_2) and (C_0, C_3).

3) Pre-load Secrets: For each secret matrix S_i, an $N \times (t+1)$ matrix $A_i = (S_i \cdot P)^T$ is computed, where $i = 1, 2, \ldots, N_S$. The matrix A_i consists of the secrets to be preloaded into the nodes in the pair of neighboring cells corresponding to S_i. Specifically, for node n_j in the pair of neighboring cells corresponding to S_i, the j-th row of A_i is preloaded into n_j's memory space.

Obviously, each secret matrix S_i is used by $2N_c$ nodes in the corresponding pair of neighboring cells. To guarantee the secrecy of S_i in the $2N_c$ nodes, the $(2N_c-2)$-secure Blom scheme should be used, which means $t = 2N_c - 2$ should be chosen and the size of S_i is $(2N_c - 1) \times (2N_c - 1)$.

Because each triangle cell has three neighboring cells, each of which has a common boundary with the cell, for the nodes in each cell, 3 secret matrices S_{i_1}, S_{i_2}, and S_{i_3} are used. Thus each node is preloaded with 3 rows from 3 different matrices A_{i_1}, A_{i_2}, and A_{i_3}, respectively. The memory cost for secrets per node is $3(2N_c - 1)$. It is to be shown later that this memory cost can be further reduced.

D. Post-deployment Phase

The secrets preloaded in the pre-deployment phase can facilitate the shared key establishment between neighboring nodes during the post-deployment phase. Generally, each node can calculate direct shared keys with some of its neighbors and negotiate indirect shared keys with the other neighbors through a secure multi-hop path.

1) Direct Keys: After deployment, each sensor node exchanges its node ID and the indices of the secret matrices used to derive secrets for the node with its neighboring nodes. If two neighboring nodes find they have the secrets from the same secret matrix, they can calculate a direct shared key without further interaction between them.

Suppose the neighboring nodes n_i and n_j have the secrets from the secret matrix S_c, which means n_i has the i-th row $A_c(i)$ of the matrix A_c and n_j has the j-th row $A_c(j)$ of the matrix A_c, where $A_c = (S_c \cdot P)^T$. Node n_i can recover the j-th column $P(j)$ of the public matrix P using the ID of node n_j, and node n_j can also recover the i-th column $P(i)$ of P. Hence, a direct shared key between node n_i and node n_j can be calculated as

$$K_{ij} = A_c(i) \cdot P(j) = A_c(j) \cdot P(i) = K_{ji} \ . \tag{3}$$

Since the ID of each node is unique, the direct shared key is also unique to the pair of neighboring nodes. This property is particularly useful for secure communications in that it may not only provide encryption services, but also provide authentication service. As an example, suppose two neighboring nodes n_i and n_j establish a direct shared key K_{ij} by following the above procedure, then they can achieve mutual authentication through the normal *challenge-response* method [17] based on K_{ij}. However, conventional schemes like [9], [10], [13] are difficult to provide the authentication service because of the reuse of keys in many nodes.

Each node has 3 rows from 3 different matrices A_{i_1}, A_{i_2}, and A_{i_3} respectively, but not all the 3 rows are involved in the establishment of direct shared keys. After the establishment of direct shared keys, the unused rows can be removed to save memory resources. The used rows are kept in nodes' memories and may be used to establish direct shared keys with new sensor nodes added in the future.

2) Indirect Keys: After exchange of its node ID and the indices of the secret matrices used to derive secrets for it, a node may find a neighboring node does not have the secrets from the same secret matrices as it uses. Thus they can not calculate a direct shared key. In this case, they can rely on a secure multi-hop path between them to negotiate an indirect shared key. Suppose there is a path consisting of nodes $n_1, n_2, \ldots, n_i$ between node n_a and n_b. Each pair of neighboring nodes along the path has a direct shared key. Because each hop along the path is secure, it is safe to exchange an indirect key between n_a and n_b with the help of the intermediate nodes. The exchanged indirect key may be exposed if one of the nodes along the path is compromised. However, according to [9], [11], a secure path between two neighboring nodes usually consists of 2 or 3 hops, thus the probability of indirect key exposal is small. If strong security is desired, multi-path routing approaches such as *SPREAD* [18] can be applied to securely exchange the indirect keys. For the lack of space, The further investigation on this issue is left to the extension of this paper.

However, because the node deployment knowledge is used, the nodes holding secrets from the same secret matrices are located close to each other. Apparently, each node can calculate direct keys with almost all its neighbors and does not need to spend too much energy on the indirect key establishment through multi-hop routing. Hence, our scheme can save a lot of communication overhead compared with previous key pre-distribution schemes [9]–[12].

3) Node Revocation: During the operation of the network, it is possible that some nodes are compromised by adversaries. Hence the memberships of compromised nodes need to be canceled and their keys need to be revoked. It can be easily achieved when other nodes remove the corresponding keys out of their memory.

4) Node Addition: When more nodes are destroyed, some holes may exist in the network. In the case, some new nodes need to be deployed to recover the holes. According to the cell where the new nodes are to be deployed, the new nodes are preloaded with the secrets derived from the corresponding secret matrices. After deployment, the new nodes can establish direct keys and indirect keys with surrounding nodes.

Analysis and Evaluation

Here we carry out some performance evaluation of the proposed scheme on the resilience to the node compromise attack, memory cost, and local secure connectivity. We compare our scheme with three schemes, i.e., E-G [9], MSK [11], and PIKE [12].

A. Metrics

1) Resilience to the Node Compromise Attack: Due to cost constraints, it is unrealistic and uneconomical to employ tamper-resistant hardware to secure the cryptographic materials in each individual node. Hence adversaries may capture any node and compromise the secrets stored in the node. Furthermore, adversaries can use the compromised secrets to derive more secrets stored in other normal nodes. It means that the node compromise attack is unavoidable. What we can do is to reduce the impact on other normal nodes as much as possible when some nodes are compromised. The *additional key exposal probability* is used here to evaluate the resilience to the node compromise attack. Specifically, the probability that the keys stored in normal nodes are exposed should be as small as possible when some nodes are compromised.

2) Memory Cost: We will calculate how many memory units per node are necessary for key establishment, where each memory unit can accommodate a cryptographic key in conventional schemes or a matrix element. Due to the resource constraint of sensor nodes, the small memory cost is desirable.

3) Local Secure Connectivity: Local secure connectivity is the probability that two neighboring nodes establish a direct key. It also reflects the portion of neighbors a node can establish direct keys. In sensor networks, high local secure connectivity is desirable because it means each node does not need to spend too much energy on the establishment of indirect keys with neighbors through multi-hop routing.

B. Resilience To the Node Compromise Attack

Our scheme applies Blom's scheme in each pair of neighboring cells. Blom's scheme has the t-secure property that the collusion of less than $t+1$ nodes can not reveal the secret matrix S, which means the keys in normal nodes are secure even if up to t nodes are compromised. In our scheme, each secret matrix S_i is used in a pair of neighboring cells including $2N_c$ nodes. By setting $t = 2N_c - 2$, we obtain the $(2N_c - 2)$-secure property in each pair of neighboring cells, which means that even though up to $2N_c - 2$ nodes are compromised the remaining 2 nodes in the neighboring cells can still perform secure communications. Hence, in our scheme when an adversary compromises some nodes, he/she only knows the keys in the compromised nodes but can not derive the keys stored in other non-compromised nodes, which means the additional key exposal probability is zero. Thus, our scheme has perfect resilience to the node compromise attack.

PIKE can also achieve the same additional direct key exposal probability because of the pre-distribution of unique direct keys.

In E-G scheme, every time an adversary compromises one more nodes, he/she obtains more keys in the global key pool. The additional key exposal probability of E-G may be calculated as [10], [13]

$$P_c = 1 - \left(1 - \frac{M}{R}\right)^x , \tag{4}$$

where each node randomly selects a key subset of size M from a global key set of size R and the number of compromised nodes is x.

In MSK scheme, the additional key exposal probability can be calculated as [11]

$$P_c = \sum_{j=\lambda+1}^{x} \binom{x}{j} \left(\frac{\tau}{\omega}\right)^j \left(1 - \frac{\tau}{\omega}\right)^{x-j} , \tag{5}$$

where each node has τ spaces from ω spaces, each space is exposed if more than λ nodes are compromised, and x is the total number of compromised nodes.

An example: Suppose 1000 nodes are deployed in an area $1000 \times 1000m^2$. Suppose each node has a memory size of 200 units for cryptographic materials and each memory unit can accommodate a cryptographic key or a polynomial coefficient. The global key pool of E-G is set 100000. To maintain the same secure connectivity as E-G, in MSK we set $\tau = 2$ and $\omega = 10$ and $\lambda = 99$. Fig. 2 gives the additional direct key exposal probability according to the fraction of compromised nodes. We can see, our scheme as well as PIKE outperforms E-G and MSK schemes with the zero probability of the additional direct key exposal.

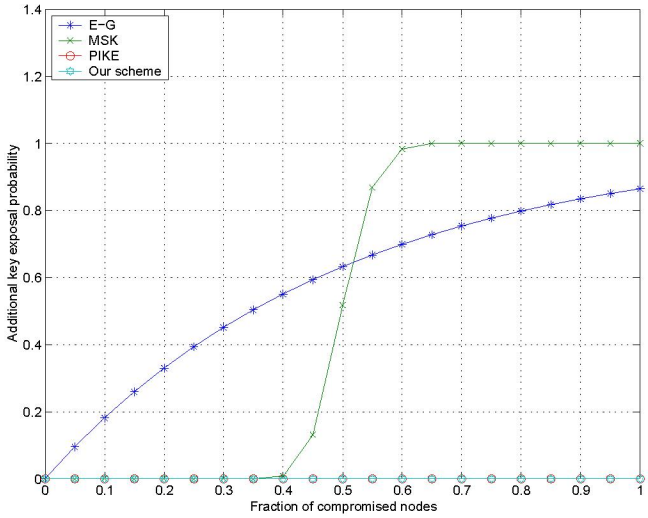

Figure 2. Probability of direct key exposal v.s. Fractional of compromised nodes.

C. Memory Cost

Each node is preloaded with 3 rows from 3 different matrices A_{i_1}, A_{i_2}, and A_{i_3} and each row has $t+1$ elements. To guarantee the secrecy of each secret matrix S_i, the value of t is set to $2N_c - 2$ where N_c is the number of nodes in one cell. Hence the memory cost for secrets per node is $3(2N_c - 1)$. However, as stated in Section III, unused rows can be removed when each node has established shared keys with its neighbors. The real memory cost is less than $3(2N_c - 1)$.

In E-G and MSK schemes, to maintain a certain local secure connectivity, which is the probability that two neighboring nodes can establish a direct shared key, the number of keys or spaces can not be too small. However, large number of keys or spaces means the adversary can obtain more secrets each time he/she compromises one more node. The contradictive memory requirements make it difficult to optimize both security and local secure connectivity given fixed memory resource. In PIKE the memory cost is $2(\sqrt{N} - 1)$ per node, where N is the total number of nodes in the network. It increases with the network size. A merit of our scheme is that the memory cost is unrelated with local secure connectivity, thus it is easy to achieve high level security. The usage of deployment also brings high local secure connectivity. Besides, our scheme is scalable in that the number of cells can increase while the memory cost per node is fixed because the cell size is fixed.

D. Local Secure Connectivity

Every node can calculate direct keys with some neighbors, and establish indirect keys with other neighbors through multi-hop routing. If a node has high probability to calculate direct keys, it can save energy on the establishment of indirect keys through multi-hop routing. Hence, high local secure connectivity, which

TABLE I. Local secure connectivity of different schemes

Schemes	Local Secure Connectivity
E-G	0.4
MSK	0.38
PIKE	0.06
Our Scheme	0.99

is the probability of establishment of direct keys, is desirable in sensor networks.

Suppose nodes are uniformly deployed in each cell. The local secure connectivity can be calculated as the ratio of the node secure coverage to the node coverage, where the secure coverage is a portion of the node coverage and the node can direct keys with the neighbors in its secure coverage. Suppose the side length of each cell is $\sqrt{3}D$, node radio radius is R. Due to the symmetry of square cell, we only consider the areas a and b in the polar coordinate plane (Fig. 3). When a node is located at (r, θ), its secure coverage $A(r, \theta)$ can be calculated as

$$A(r,\theta) = \tag{6}$$

$$\begin{cases} \pi R^2 \ , & \text{when } \frac{R}{\cos\theta} < r \le \frac{\sqrt{3}D}{2\cos(\frac{\pi}{6}-\theta)}, 0 \le \theta \le \frac{\pi}{6} \\ r\cos\theta\sqrt{R^2-(r\cos\theta)^2} + (\pi - \arccos\frac{r\cos\theta}{R})R^2 \ , & \\ \text{when } 0 \le r \le \frac{R}{\cos\theta}, 0 \le \theta \le \frac{\pi}{6} \ , & \end{cases} \tag{7}$$

The local secure connectivity can be calculated as

$$C = \frac{8}{\sqrt{3}\pi R^2 D^2} \int_0^{\frac{\pi}{6}} \int_0^{\frac{\sqrt{3}D}{2\cos(\frac{\pi}{6}-\theta)}} A(r,\theta)\, r dr d\theta \ . \tag{8}$$

In E-G scheme, each node randomly selects M keys from S keys, thus the local secure connectivity of E-G is

$$C = 1 - \frac{\binom{S-M}{M}}{\binom{S}{M}} \approx 1 - \left(1 - \frac{M}{S}\right)^M \approx \frac{M^2}{S} \ , \tag{9}$$

where $S \gg M$. In MSK scheme, each node randomly selects τ spaces from ω spaces, thus the local secure connectivity of MSK is

$$C = 1 - \frac{\binom{\omega-\tau}{\tau}}{\binom{\omega}{\tau}} \ . \tag{10}$$

In PIKE, each node keep unique pairwise keys with $2(\sqrt{N}-1)$ nodes, thus the local secure connectivity of PIKE is $2(\sqrt{N}-1)/N$.

Suppose the same configuration parameters in Section IV-B is used here and also suppose node radio radius is $25m$, the side length of each cell is $100\sqrt{3}m$. The local secure connectivity of different schemes is given in TABLE I. We can see due to the usage of deployment information, our scheme has very high local secure connectivity, which means each node can establish direct keys with almost all its neighbors, thus can save a lot of energy on the establishment of indirect keys through multi-hop paths. This is a desired property for resource constrained sensor networks.

CONCLUSION

In this paper we proposed a novel scheme for the key establishment in sensor networks. By using Blom's scheme our scheme can achieve perfect resilience to the node compromise attack with low memory cost. Based on the triangle grid deployment model, our scheme can also achieve high local secure connectivity, with is unrelated to memory cost. Compared with conventional schemes, our scheme is more secure and more efficient.

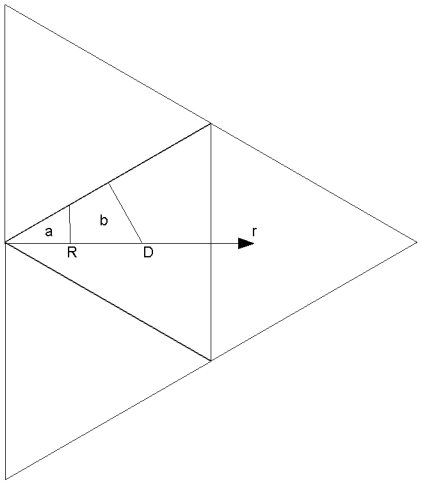

Figure 3. Node coverage in one cell.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Communication Magazine, vol. 40, no. 8, pp. 102-114, August 2002.

[2] J. M. Kahn, R. H. Katz and K. S. J. Pister, "Next century challenges: Mobile networking for Smart Dust," in ACM/IEEE International Conference on Mobile Computing and Networking(MobiCom 99), p. 217-278, August 1999.

[3] G. J. Pottie, W. J. Kaiser, "Wireless integrated network sensors," in Communications of the ACM 43(5)(2000)551-558.

[4] H. T. Kung, and D. Vlah, "Efficient location tracking using sensor networks," in IEEE Wireless Communications and Networking Conference(WCNC), March, 2003.

[5] R. Brooks, P. Ramanathan, and A. Sayeed, "Distributed target classification and tracking in sensor networks," in Proceedings of the IEEE, vol.91, no.8, pp.1163-1171, 2003.

[6] A. Wood and J. Stankovic, "Denial of service in sensor networks," IEEE Computer, pp. 54-62, October 2002.

[7] Chris Karlof, David Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," First IEEE International Workshop on Sensor Network Protocols and Applications, May 2003.

[8] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: Security protocols for sensor networks," Wireless Networks, vol. 8, pp. 521-534, September 2002.

[9] L. Eschenauer and V. Gligor, "A key management scheme for distributed sensor networks," in ACM CCS2002, Washington D.C., 2002.

[10] Haowen Chan, Adrian Perrig, and Dawn Song, "Random key predistribution schemes for sensor networks," in Proceedings of the 2003 IEEE Symposium on Security and Privacy, p.197, May 11-14, 2003.

[11] W. Du, J. Deng, Y. S. Han, and P. K.Varshney, "A pairwise key predistribution scheme for wireless sensor networks," in CCS'03, Washington, DC, October 27-30, 2003.

[12] H. Chan and A. Perrig, "Pike: peer intermediaries for key establishment in sensor networks," in *IEEE INFOCOM'05*, March, 2005.

[13] W. Du, J. Deng, Y. S. Han, S. Chen and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in the IEEE INFOCOM 2004, Hong Kong, March 2004.

[14] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanism for large-scale distributed sensor networks," in ACM CCS'03, Washington, DC, October 27-31, 2003.

[15] R. Blom, "An optimal class of symmetric key generation systems," in *Proc. of EUROCRYPT '84*, pages 335-338, 1985.

[16] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correction Codes*, North-Holland, New York, 1977.

[17] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, ISBN: 0-8493-8523-7, October 1996.

[18] W. Lou, W. Liu and Y. Fang, "SPREAD: Enhancing data confidentiality in mobile ad hoc networks," in *IEEE INFOCOM'04*, HongKong, China, Mar 2004.