A Location-Aware Scheme for Key Establishment in Wireless Sensor Networks

Ashok Kumar Das, Abhijit Das
{akdas,abhij}@cse.iitkgp.ernet.in
Surjyakanta Mohapatra
surjyakanta@gmail.com
Srihari Vavilapalli
hvpalli@yahoo.com
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur 721 302, India

***Abstract*— Key establishment in sensor networks is a challenging problem since asymmetric cryptosystems are unsuitable for use in resource constrained sensor nodes, and also because the nodes are vulnerable to physical capture. The key establishment schemes proposed recently in the literature are known to yield higher connectivity and better resilience against node captures if some prior knowledge of deployment locations of the nodes are available. In this paper, a location-aware key predistribution scheme, called the *t-neighborhood scheme*, is proposed, which provides unconditional security and works for any deployment topology. Our scheme provides better trade-off among connectivity, security and generality compared to the schemes published so far, and can be adapted to support addition of nodes after the initial deployment.**

I. Introduction

Sensor networks have been widely deployed in a variety of applications ranging from military operations like enemy base monitoring and target tracking to scientific and industrial operations like environmental and wild-life explorations and monitoring of nuclear power plants. Typically thousands of sensor nodes are deployed in the area of operation, that set up communication with each other and exchange information among each other. In certain specific situations, typically for military and medical applications, one requires secrecy and confidentiality during message transfer. A survey on sensor network can be found in [1].

An important challenge for securing communication in sensor networks is the design of protocols to establish cryptographically secure communication links among the sensor nodes. This protocol is popularly called the *bootstrapping protocol* [2]. One should also allow fresh nodes to join the network after the initial deployment, for example, to replenish dead, damaged and captured nodes. The biggest technical difficulty of bootstrapping arises due to the limitations of computing and communication resources in sensor nodes such as processing power, available memory and communication range.

Some techniques [3], [2], [4], [5], [6], [7] are already proposed in the literature for solving the bootstrapping problem. Eschenauer and Gligor propose the first basic protocol called the EG scheme [3]. It is based on the predistribution of keys randomly selected from a big key pool to the key rings of the sensor nodes. Chan et al. propose three variations of the EG scheme [2]. Liu and Ning's polynomial-pool based key predistribution scheme [4] and the matrix-based key predistribution scheme [7] due to Du et al. improve security considerably.

A key predistribution scheme is said to be *location-aware* if it works with prior knowledge (perhaps approximate) of the deployment area and the deployment locations of the sensor nodes. This knowledge can be effectively exploited to tune the key predistribution schemes so as to achieve better connectivity and higher resilience against node capture. Some published location-aware schemes include the closest pairwise scheme [7], the bivariate polynomial-pool based scheme [7] and the location-aware EG scheme [5]. In this paper, we propose an improved location-aware polynomial-pool based scheme.

Section II gives an overview of some existing location-aware schemes. In Section III, we propose our scheme called the *t-neighborhood scheme*. This scheme is based on the polynomial-pool scheme and is designed to guarantee unconditional security and to cater to arbitary deployment configuration. We also provide a theoretical analysis of our scheme. In Sections IV and V, we report our simulation results and compare the performance of the *t-neighborhood scheme* with that of the closest pairwise scheme and of the bivariate polynomial-pool based scheme. We conclude the paper in Section VI.

II. Existing Location-Aware Key Predistribution Schemes

In this section we provide a brief overview of the two location-aware schemes described in [7]. The location-aware EG scheme proposed in [5] has a somewhat different flavor in that it is based on a particular deployment topology. This limitation makes this scheme not directly comparable with our scheme and hence a study of this scheme is omitted in this paper.

A. The Closest Pairwise Key Distribution Scheme (CPKS)

Let there be n sensor nodes in a network with each node being capable of storing m symmetric keys. The expected deployment location of each node is provided to the key set-up server.

Key predistribution phase: For each sensor node u in the network, the server determines a set S of m other nodes whose expected locations of deployment are closest to that of u. For every node v in S, for which a pairwise key between u and v has not already been established, a new random key k_{uv} is generated. The key-plus-id combination (k_{uv}, v) is loaded to u's key ring, whereas the pair (k_{uv}, u) is loaded to v's key ring.

Direct key establishment (shared key discovery) phase: After the deployment of the sensor nodes, two nodes u and v can establish a secure communication link, if they share a predistributed pairwise key. To identify a common key is trivial, because each pairwise key in a particular node is accompanied by the id of the other node holding the key.

The network connectivity and security against node compromise of CPKS are as follows:

1) Connectivity: Connectivity of CPKS depends upon the deployment error. The benefits of location-awareness decrease as the error between expected and actual deployment locations increases. For sufficiently large errors, CPKS essentially degrades to the random pairwise keys scheme [2].

2) Security: Each predistributed pairwise key is randomly generated. Thus, no matter how many nodes are captured, the pairwise keys between uncompromised sensor nodes remain secure. Since the size of the total key space is typically much bigger than the size of the network, it is assumed that random pairwise keys are not repeated (birthday paradox [8]). In this sense, CPKS provides unconditional security.

B. The Location-Aware Key Distribution using Bivariate Polynomials (Bivariate Poly-Pool Scheme)

The deployment region is partitioned into a two-dimensional array of rectangular cells. Let the partition consist of R rows and C columns. The cell located at the i-th row and the j-th column is denoted by $C_{i,j}$. The neighbors of the cell $C_{i,j}$ are taken to be the four adjacent cells: $C_{i-1,j}, C_{i+1,j}, C_{i,j-1}, C_{i,j+1}$.

The key set-up server chooses RC random symmetric t-degree bivariate polynomials $f_{i,j}(X,Y) \in F_q[X,Y]$, $i = 1, 2, \ldots, R, j = 1, 2, \ldots, C$, where F_q is a finite field with q large enough to accomodate a cryptographic key. Let the expected deployment location of node u lie in the cell $C_{i,j}$ called the *home cell* of u. The key ring of u is loaded with the shares (evaluated at u) of the five polynomials corresponding to the home cell and the four neighboring cells. That is, u gets the five polynomial shares: $f_{i,j}(X,u), f_{i-1,j}(X,u), f_{i+1,j}(X,u), f_{i,j-1}(X,u), f_{i,j+1}(X,u)$. The key set-up server also stores in u's memory the id (i,j) of its home cell.

In the direct key establishment phase, each node u broadcasts the id (i,j) of its home cell (or some messages encrypted by potential pairwise keys). Those physical neighbors of u, whose home cells are either the same as or neighboring to that of u, can establish pairwise keys with u.

The network connectivity and security against node compromise of this scheme are as follows:

1) Connectivity: Analogous to CPKS, connectivity of the bivariate poly-pool scheme depends on the deployment error. Larger error leads to poorer connectivity.

2) Security: As long as no more than t polynomial shares of a bivariate polynomial are disclosed, an attacker knows nothing about the non-compromised pairwise keys established using this polynomial. Thus, the security of this scheme depends on the average number of nodes sharing the same polynomial, or equivalently on the number of nodes that are expected to be located in each cell and its four adjacent cells. If that number is larger than t, the bivariate poly-pool scheme is not unconditionally secure.

III. The Location-Aware t-neighborhood Scheme

Let us now introduce our location-aware t-neighborhood scheme. Our scheme is based on shares of symmetric bivariate polynomials over finite fields. The basic goals that this scheme tends to achieve are as follows:

- *Unconditional security:* Let t be the degree of each bivariate polynomial to be used. No more than t shares of a polynomial are distributed among the nodes.
- *Generality:* Our scheme works for any deployment configuration. There is no need to assume a rectangular area of deployment (as in [7]) or a rectangular grid-based deployment (as in [5]).

A. Key Predistribution

The key set-up server generates a pool of s randomly generated symmetric bivariate polynomials of degree t over a finite field F_q. For each polynomial $f(x,y)$ in the pool only t shares of $f(x,y)$ are distributed to the nodes as per the following rules. We assume that each node is capable of storing a maximum of s' shares.

Select a node u randomly that has less than s' shares in its memory and distribute a share of $f(x,y)$ to u. The remaining $t-1$ shares of $f(x,y)$ are distributed among the expected neighbors of u. The left over shares, if any, are distributed to the neighbors of neighbors of u, and so on. A share is never given to a node which has already been loaded fully (i.e., which has already been distributed s' shares).

B. Shared Key Discovery

After deployment, two nodes u and v that are in communication ranges of one another exchange their ids as well as the ids of the polynomials whose shares they possess. If they have the share of a common polynomial $f(x,y)$, they calculate the pairwise key $f(u,v) = f(v,u)$ for use in future secure communication.

C. Dynamic Node Addition

To dynamically add new nodes to the network, it makes no sense to use shares of polynomials which are not already employed during initial key set-up, since new polynomials do not allow the new nodes to establish any communication link with the nodes previously deployed. It is also not a good idea to give the shares of polynomials already used in the network, as it will make the number of shares of some polynomials exceed t thereby violating unconditional security. Instead of distributing t shares in the key predistribution phase, less than t shares of the polynomials can be distributed. Suppose that h shares of a certain polynomial are distributed in the first phase. Some of the remaining $(t-h)$ shares are to be distributed to the new nodes according to their expected locations. However, since t is a finite number (typically ≤ 200), this method imposes a restriction on the number of times new nodes can be added.

D. Theoretical Analysis of the t-neighborhood Scheme

For the sake of simplicity, we assume that the target field is two-dimensional, so that every point in that region is expressed by two co-ordinates x and y. Assume that u is a sensor node whose expected location is (u_x, u_y) whereas its actual location is (u'_x, u'_y). This corresponds to a deployment error of $e_u = (u'_x - u_x, u'_y - u_y)$. The actual location (or equivalently the error e_u) can be modeled as a continuous random variable that can assume values in R^2. The probability density function $f_u(u'_x, u'_y)$ of (u'_x, u'_y) characterizes the pattern of deployment error. As in [7] we assume that (u'_x, u'_y) is uniformly distributed within a circle with center at (u_x, u_y) and radius e called the *maximum deployment error*. We have:

$$f_u(u'_x, u'_y) = \begin{cases} \frac{1}{\pi e^2} & \text{if } (u'_x - u_x)^2 + (u'_y - u_y)^2 \leq e^2 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Another strategy (see [5]) is to model (u'_x, u'_y) as a random variable following the two-dimensional normal (Gaussian) distribution with mean (u_x, u_y) and variance σ^2. The corresponding probability density function is:

$$f_u(u'_x, u'_y) = \frac{1}{2\pi\sigma^2} e^{-[(u'_x-u_x)^2+(u'_y-u_y)^2]/(2\sigma^2)}. \tag{2}$$

However, it is quite clumsy to work with this distribution and so we will stick to the uniform distribution.

Two nodes are called *physical neighbors* if they lie in each other's communication range. They are called *key neighbors* if they possess shares of a common polynomial. They are called *direct neighbors* if they are both physical and key neighbors.

Let u and v be two deployed nodes. Assume that each node has a communication range ρ and that the different nodes are deployed independently i.e., (u'_x, u'_y) and (v'_x, v'_y) are independent random variables.

The probability that u and v are in each other's communication range can be calculated by

$$p(u,v) = \iiiint_C f_u(u'_x, u'_y)\, f_v(v'_x, v'_y)\, du'_x\, du'_y\, dv'_x\, dv'_y \tag{3}$$

where C is the region $(u'_x - v'_x)^2 + (u'_y - v'_y)^2 \leq \rho^2$. This expression makes use of the fact that u and v are independently deployed.

Let d be the number of physical neighbors of u. Assume that the key neighbors of u are uniformly distributed in a circle of radius ρ'. Since u shares pairwise keys with at most $s'(t-1)$ nodes, the expected value of ρ' is $\rho' = \rho \times \sqrt{\frac{s'(t-1)}{d}}$. If v is a key neighbor of u, the probability that v lies in the physical neighborhood of u is given by

$$p(u) = \frac{1}{\pi\rho'^2} \iint_{C'} p(u,v)\, dx\, dy \tag{4}$$

where C' is the region $(x-u_x)^2 + (y-u_y)^2 \leq \rho'^2$. Since u is expected to have $s'(t-1) \times p(u)$ direct neighbors, the probability that u can establish a pairwise key with a physical neighbor is given by

$$p = p(u) \times \frac{s'(t-1)}{d} = p(u) \times \lambda \tag{5}$$

where $\lambda = \frac{s'(t-1)}{d}$. The probability p (averaged over all nodes u) measures the local connectivity of the network.

We use the communication range ρ as the basic unit of distance measurement, i.e., $\rho = 1$. One can compute the probability p for the density function given by Equation(1) and establish that $p \approx 1$ for small deployment errors.

Since each node holds at most s' polynomial shares, each key ring requires a space of $s'(t+1)\log q$ bits, which is the same as is needed for storing $m = s'(t+1)$ symmetric keys.

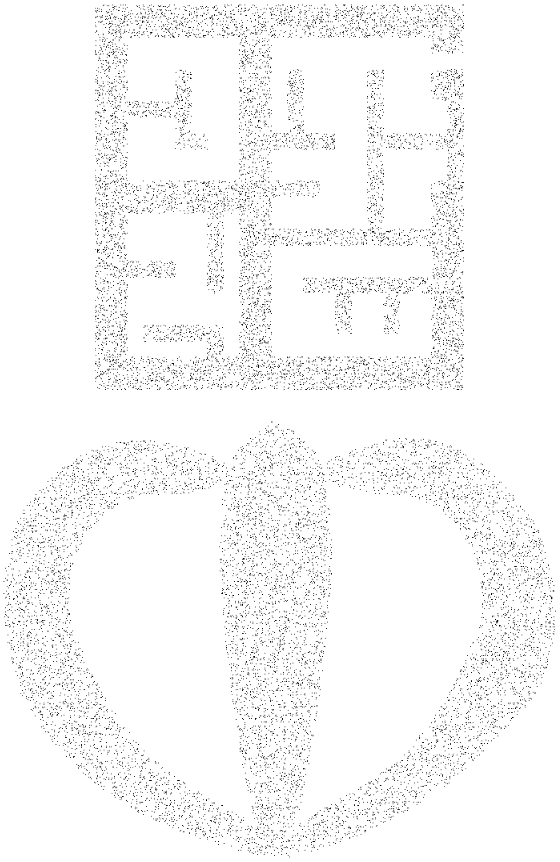

Fig. 1. Two random deployment models

The value of t depends on the value of s'. For example, we have $t = 39$ when $s' = 5$ and $m = 200$.

IV. Simulation Results

For the simulation of our scheme we have considered several arbitary deployment models of which two are shown in Figure 1. Each dot in the figures represents the expected deployment location of a sensor node.

We have taken the following parameters for the simulation:

- Number of nodes in the network, $n = 10000$.
- Average number of neighbors of a node, $d \leq 100$.
- Size of the key ring of a node (in number of symmetric keys), $m = 200$.
- Size s of the polynomial pool is chosen so that the maximum supported network size becomes $n = \frac{(t+1)s}{s'}$. For example, $s = 1250$ when $n = 10000$, $s' = 5$ and $t = 39$.

Figure 2 shows that the analytical and simulation results of direct network connectivity tally closely for the t-neighborhood scheme with the parameter values $d = 80$, $s = 1250, s' = 5$, and $t = 39$. This figure indicates that when the error e increases, the connectivity of the network degrades. The simulation results of direct network connectivity for the t-neighborhood scheme are plotted in Figure 3 for different values of s'. This figure also shows that when the error e increases, the connectivity of the network degrades.

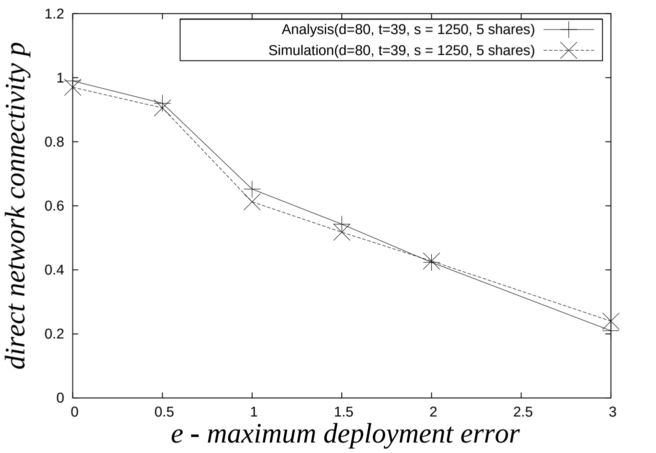

Fig. 2. Analysis v.s. simulation results of direct connectivity of the t-neighborhood scheme, with $d = 80$, $s = 1250$, $s' = 5$, and $t = 39$.

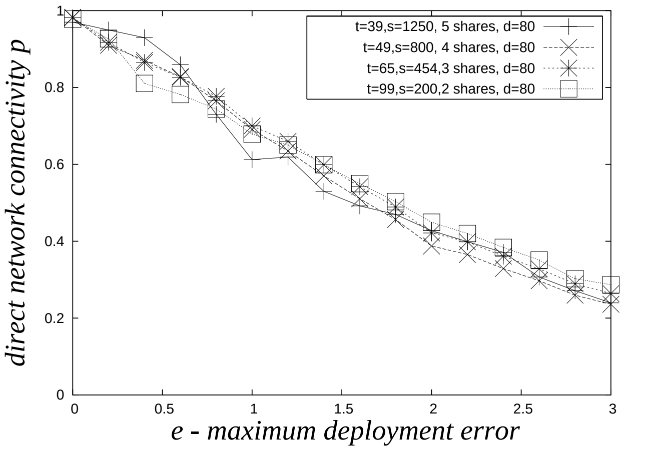

Fig. 3. Direct network connectivity of the t-neighborhood scheme with $s = 200, 454, 800, 1250$, $s' = 2, 3, 4, 5$, and $t = 99, 65, 49, 39$.

V. Comparison with Previous Schemes

The proposed scheme is compared with two other schemes, the closest pairwise scheme (CPKS) and the bivariate poly-pool scheme. The simulation results for the three schemes are shown in Figure 4. The performance of our scheme is slightly poorer than the other schemes. This degradation of performance of our scheme can be justified as an acceptable trade-off between connectivity and certain important criteria considered below.

The t-neighborhood scheme distributes polynomial shares to a randomly selected node and to its neighbors. So, it works for any topology. In case of CPKS, if nodes u and v belong to two uneven distribution zones, v may be among of u's closest neighbors but u may not be v's [7]. Our scheme eliminates this problem by distributing shares of each polynomial centered around a chosen node.

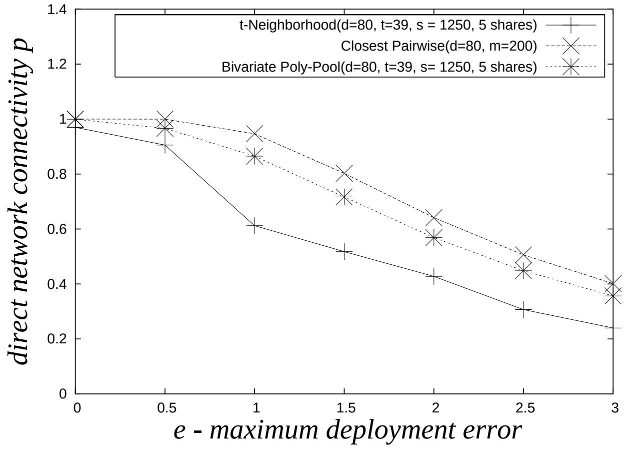

Fig. 4. Network connectivity of the t-neighborhood scheme, the bivariate poly-pool scheme, and the closest pairwise scheme.

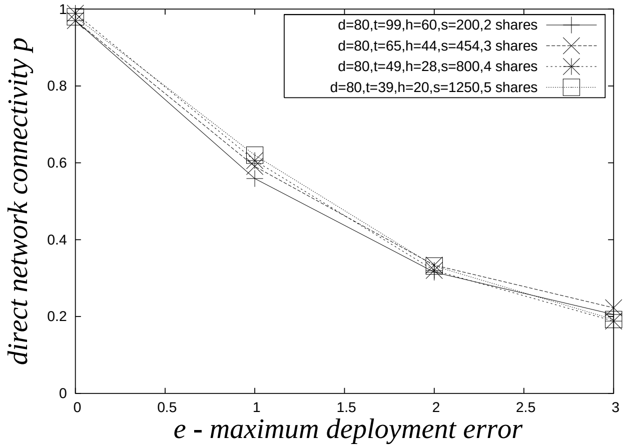

Fig. 5. Direct network connectivity of the t-neighborhood scheme for dynamic node addition, with $s = 200, 454, 800, 1250$, $s' = 2, 3, 4, 5$, $t = 99, 65, 49, 39$, and $h = 60, 44, 28, 20$.

In CPKS dynamic node addition becomes difficult. A new deployed node has to be distributed some keys already present in the network. Repeated use of the keys may make the network vulnerable to node captures. On the other hand, for the t-neighborhood scheme dynamic addition of nodes does not affect the security but degrades the connectivity only nominally, as demonstrated in Figure 5.

The t-neighborhood scheme is unconditionally secure. This means that no matter how many nodes are captured, the remaining uncaptured nodes in the network can communicate with 100% secrecy. CPKS also possesses this security guarantee. In case of the bivariate poly-pool scheme, the average number of shares of a single polynomial may be more than t, making this scheme less secure than the t-neighborhood scheme.

VI. Conclusion

In this paper, we propose a location-based scheme called the *t-neighborhood scheme* which takes advantages of prior knowledge of deployment locations of the sensor nodes. Our scheme provides better security compared to the previous schemes, ensures reasonable connectivity and supports dynamic node addition.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40(8), August 2002.

[2] H. Chan, A. Perrig, and D. Song, “Random key predistribution for sensor networks,” in *IEEE Symposium on Security and Privacy*, Berkely, California, 2003, pp. 197–213.

[3] L. Eschenauer and V. D. Gligor, “A key management scheme for distributed sensor networks,” in *9th ACM Conference on Computer and Communication Security*, November 2002, pp. 41–47.

[4] D. Liu and P. Ning, “Establishing pairwise keys in distributed sensor networks,” in *Proceedings of 10th ACM Conference on Computer and Communications Security(CCS)*, Washington DC, Oct 27-31 2003, pp. 52–61.

[5] 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 *23rd Conference of the IEEE Communications Society (Infocom’04)*, Hong Kong, China, March 21-25 2004.

[6] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, “A pairwise key predistribution scheme for wireless sensor networks,” in *ACM Conference on Computer and Communications Security (CCS’03)*, Washington DC, USA, October 27-31 2003, pp. 42–51.

[7] D. Liu and P. Ning, “Location-based pairwise key establishments for static sensor networks,” in *Proceedings of 1st ACM Workshop on security of ad-hoc and sensor networks*, Fairfax, Virginia, 2003, pp. 72–82.

[8] W. Stallings, *Cryptography and Network Security: Principles and Practices*, 3rd ed. Prentice Hall, 2003.