LLK: A Link-Layer Key Establishment Scheme for Wireless Sensor Networks

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*— The establishment of link-layer keys between neighboring nodes is a fundamental issue in securing sensor network communications. Most of existing solutions are key predistribution schemes which rely on sensor nodes to broadcast hundreds of or even thousands of pre-loaded key IDs to find pairwise keys between neighboring nodes. The shortcomings include poor resilience against node compromise, low network connectivity, large communication overhead, etc. This paper presents a novel location-based link-layer key establishment scheme, in which a hexagonal-grid-based deployment model and a polynomial-based key establishment model are combined for the first time to establish a link-layer key between two neighboring nodes. Compared with conventional proposals, our scheme features much lower communication overhead and memory requirements while still maintaining high network connectivity and network resilience against node compromise.**

I. 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. Sensor networks can provide fine-granular sensing and intelligent computation and control services in a lot of applications, e.g., medical care, emergency response, environmental pollution monitoring, etc. Sensor networks, however, are vulnerable to malicious attacks in unattended and hostile environments such as battlefield surveillance and homeland security monitoring. For instance, enemies can easily eavesdrop messages transmitted over the air between sensor nodes, or disable the whole sensor network by launching physical attacks to sensor nodes or logical attacks to communication protocols [1], [2]. Under such circumstances, secure mechanisms are indispensable for guaranteeing the proper operation of sensor networks.

The basic secure mechanism is the use of cryptography. For example, by encrypting messages, we can prevent message eavesdropping, while by authenticating messages, we can prevent message alteration or falsification. There are two basic encryption schemes, namely, *transport-layer encryption* and *link-layer encryption*. In the former scheme, the source node sends to the destination a message encrypted with a key uniquely shared with the destination (called *transport-layer key* hereafter) and any intermediate node on the forwarding path can not decrypt the message for the lack of the transport-layer key. In the latter scheme, each pair of neighboring nodes shares a unique key called a *link-layer key*. When a packet is forwarded from the source to the destination, each intermediate node will decrypt the packet with the link-layer key shared with the predecessor, process the packet header, re-encrypt the packet with the link-layer key shared with the successor, and forward it further. Both schemes have their own drawbacks. The transport-layer encryption scheme is vulnerable to traffic analysis attacks in that only transport-layer PDUs are encrypted so that enemies may ascertain the identities of the source and destination by simply reading IP headers and hence derive the network topology, which is undesirable in many scenarios. And the link-layer encryption scheme suffers from node compromise attacks in the sense that a single compromised intermediate node can read all the messages forwarded by it. It is, therefore, necessary to combine the two schemes together to provide a better solution to enhance the sensor network security. That is, a message is first encrypted by the source with the transport-layer key shared with the destination and then encrypted/decrypted with the link-layer keys shared between neighboring intermediate nodes in a hop-by-hop fashion. As a result, enemies cannot easily ascertain the identities of the source and the destination as before and read the message contents forwarded through compromised nodes.

Generally, a transport-layer key can be negotiated on demand between the source and the destination if the intermediate links are secure, that is, each pair of intermediate neighboring nodes has a link-layer key. Therefore, it is important to figure out an efficient way to establish link-layer keys between any two neighboring nodes.

In this paper, by utilizing node deployment information, we propose a novel location-based link-layer key establishment scheme, named *LLK*, in which a hexagonal-grid-based deployment model and a polynomial-based key establishment model are combined for the first time to establish a link-layer key between any two neighboring nodes. Compared with conventional proposals, our scheme features much lower communication overhead and memory requirements while still maintaining high network connectivity and network resilience against node compromise.

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

0-7803-8966-2/05/$20.00 (C) 2005 IEEE

The rest of the paper is organized as follows. Section II surveys the related work. Section III gives the mathematic background of the polynomial-based key establishment. Section IV describes the hexagonal-grid-based deployment model. Section V presents the details of our scheme. Section VI gives some analysis of our scheme. Section VII compares our scheme with related work. Section VIII concludes the paper and points out several future directions.

II. Related Works

Recent research has seen a growing body of work on establishing pairwise keys in sensor networks, most of which are probabilistic key pre-distribution schemes, where keying materials are pre-loaded to all the sensor nodes before deployment. These schemes may be used for link-layer key establishments. Two notable solutions include the *Eschenauer-Gligor* scheme [3] and *q-composite random key pre-distribution* scheme [4], in which each node is pre-loaded with a random subset of keys from a global key pool in such a way that any two nodes can share at least one common key [3] or q keys [4] with a certain probability. However, as pointed out in [5], both schemes are vulnerable to node compromise attack in that a small number of compromised nodes may expose a large fraction of pairwise keys between non-compromised nodes. Liu and Ning [5] proposed a set of polynomial-based key pre-distribution protocols to improve the resilience against node compromise, where a unique pairwise key can be established between any pair of neighboring nodes. Another noticeable scheme is the *multiple-space key pre-distribution* scheme proposed in [6], where each key in [3] [4] 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. The schemes [3]–[6] only guarantee that any two neighboring nodes can directly establish a pairwise key with a pre-determined probability p. As a result, each node can just establish pairwise keys shared with a portion of its neighbors, leading to unfavorable low network connectivity [1].

Very recently, Du *et al.* [7] proposed to utilize node deployment knowledge to improve the *Eschenauer-Gligor* scheme in terms of network connectivity, memory usage, and network resilience against node compromise. Their scheme assumes a group-based deployment model, in which sensor nodes are deployed in groups around their deployment points [2] and the distribution of deployment points follows a rectangular grid model. In each group, the *Eschenauer-Gligor* scheme is applied. Therefore, their scheme is still vulnerable to node compromise attacks even though it increases the cost of such attacks.

All of the above pre-distribution schemes rely on sensor nodes to broadcast hundreds of or even thousands of indices of pre-loaded keys or key spaces in order to find pairwise keys between neighboring nodes, thus leading to huge communication overhead. In addition, to guarantee a certain network connectivity, say p, each node has to store several hundreds keys or key spaces, which may greatly increase the memory costs.

Liu and Ning presented a location-based key pre-distribution scheme [8] using bivariate polynomials, in which sensors are grouped into square grids (or cells) and each cell is associated with a unique random bivariate polynomial. Each sensor is pre-loaded with the polynomial shares of its home cell, in which the sensor is supposed to locate, and four other cells adjacent to its home cell. After deployment, any two neighboring nodes can establish a pairwise key if they have shares of the same polynomial. Compared with previous schemes, their scheme only requires one node to broadcast its home cell coordinate to the other node, hence greatly decreases the communication overhead.

III. Polynomial-Based Key Distribution

Blundo *et al.* [9] proposed to use bivariate polynomials to achieve key distribution for dynamic conferences, which forms the basis of this paper. The basic idea of their scheme is outlined as follows.

To establish a pairwise key between two nodes, the key setup server first generates a random t-degree bivariate polynomial,

$$f(x,y)=\sum_{i=0}^{t}\sum_{j=0}^{t}a_{ij}x^iy^j \ , \qquad (1)$$

over a finite field $\mathbb{F}_q$, where q is a pre-determined prime that is large enough to accommodate a cryptographic key. By choosing appropriate coefficients $a_{ij}=a_{ji}$, we can have the desirable property of symmetry, i.e., $f(x,y)=f(y,x)$. Assume that each sensor node have a unique, integer-valued, nonzero ID. For a pair of sensor nodes n_i and n_j (n_i and n_j are unique node IDs), we can assign a *polynomial share* $f(n_i,y)$ to n_i and another share $f(n_j,y)$ to n_j. After deployment, both nodes need to broadcast their IDs to establish a pairwise key. Then node n_i can compute $f(n_i,n_j)$ by evaluating $f(n_i,y)$ at point n_j, and node n_j can compute $f(n_j,n_i)$ by evaluating $f(n_j,y)$ at point n_i as well. Due to the symmetry of the bivariate polynomial, the secure pairwise key between nodes n_i and n_j is established as $K_{ij}=f(n_i,n_j)=f(n_j,n_i)$.

A t-degree bivariate polynomial has a nice property of t-collusion resistance, i.e., as long as the number of nodes sharing a polynomial is no more than t, the collusion between those nodes can not expose the polynomial. So if we assign one t-degree bivariate polynomial to no more than t sensor nodes, adversaries cannot reconstruct the original polynomial from compromised nodes and thus cannot derive the pairwise keys shared between non-compromised nodes.

IV. Hexagonal-Grid-Based Deployment Model

We assume that each sensor node has a pre-determined *deployment point* where it is supposed to reside. However, the node usually cannot be exactly at the deployment point once

[1] Similar to previous proposals, we define network connectivity as the probability that any two neighboring nodes share one key.

[2] A deployment point is defined as the point location where a senor or sensor group is to be deployed.

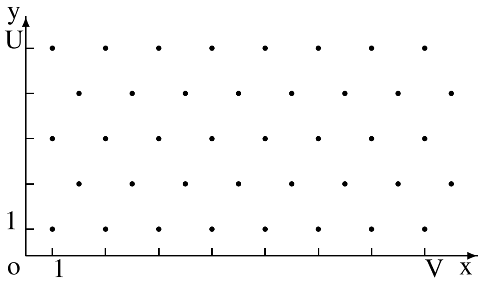

Fig. 1. Hexagonal Grid Based Deployment

TABLE I
THE ALGORITHM FOR POLYNOMIAL PRE-DISTRIBUTION.

```
For each $G_{uv}$ {
    For each neighboring group $G'_{uv}$ of $G_{uv}$ {
        If $G'_{uv}$ and $G_{uv}$ do not share a polynomial {
            Assign a $f(x, y)$ to $G'_{uv}$ and $G_{uv}$;
            Remove the $f(x, y)$ from $F$;
        }
    }
}
```

deployed. For example, when sensor nodes are deployed from an aircraft, they may fall away from their designated locations because of the wind or imprecise timings or other reasons. Such uncertain environmental factors make the real *resident point*, where a node finally resides, a variable following some probability distribution function (PDF). An example of the PDF is a two dimensional Gaussian distribution:

$$p(x, y) = \frac{1}{2\pi\sigma^2} \exp \frac{-(x^2 + y^2)}{2\sigma^2} \ , \tag{2}$$

where we assume the deployment point to be the origin of coordinates.

Schemes [3]–[6], [8] assumed $p(x, y)$ to be a uniform distribution over the deployment area. To the authors' knowledge, the Gaussian distribution was used for the first time in [7]. In this paper, we still use the Gaussian distribution because it is the most popular statistical distribution to model a center-symmetric distribution. In contrast to [7], our scheme is more robust to node compromise attacks and can greatly decrease the broadcast overhead in [7] because of the use of bivariate polynomials. Besides, it is worth pointing out that our scheme does not preclude the use of other PDFs in different application scenarios.

Deployment information has been used for link-layer key pre-distribution in previous schemes [7] [8]. But they assume that the deployment points follow a square-grid model, which may not be suitable models in many cases because of the omnidirectional coverage of each sensor node. In this paper, instead we propose the use of a hexagonal-grid-based deployment model for the first time in the literature. In particular, we assume that the deployment points in the deployment area are arranged in a *hexagonal grid* (Fig. 1) of size $U \times V$, because the hexagon is the best approximation to the shape of node coverage area, which is circle. As a result, the whole deployment area is separated into many hexagonal cells, each of which is centered with a deployment point.

V. DETAILS OF OUR SCHEME

In this section, we illustrate the details of the proposed scheme.

A. The Generation of Groups

At the beginning all G sensor nodes to be deployed are divided into $U \times V$ equal-sized, non-overlapping groups G_{uv}, for $u = 1, \ldots, U$ and $v = 1, \ldots, V$, which satisfies

$$G = \bigcup_{(u,v)=(1,1)}^{(U,V)} G_{uv} \ . \tag{3}$$

The group G_{uv} is deployed around the deployment point $d_{uv} = (x_u, y_v)$ with index (u, v). In our scheme, the number of nodes in one group is no more than $G_c = |G_{uv}| \leq \lfloor \frac{t}{2} \rfloor$. The reason will be stated later. The resident points $r_i^{uv} = (x_i^{uv}, y_i^{uv})$ of the node n_i^{uv} in group G_{uv} follows a two dimensional Gaussian distribution. When the deployment point of group G_{uv} is at $d_{uv} = (x_u, y_v)$, we have the PDF for node n_i^{uv} in group G_{uv}:

$$\begin{aligned} p^{uv}(x, y) &= \frac{1}{2\pi\sigma^2} \exp \frac{-[(x - x_u)^2 + (y - y_v)^2]}{2\sigma^2} \\ &= p(x - x_u, y - y_v) \ . \end{aligned} \tag{4}$$

In addition, each node is preloaded with the location (x, y) of the deployment point of its affiliated group.

B. The Pre-distribution of Polynomials

Before deployment, we construct a global polynomial pool F with enough t-degree bivariate polynomials. Each polynomial in F is used to derive polynomial shares for nodes in two neighboring cells. Since in the hexagonal grid model, each group has at most 6 neighboring groups, each node will carry at most 6 polynomial shares which will be used to establish link-layer keys. Recall that the number of nodes in one group is no more than $\lfloor \frac{t}{2} \rfloor$, so that each polynomial will be assigned to no more than t nodes. Hence, the adversary cannot derive the link-layer keys shared between non-compromised sensor nodes from the information stored in the compromised sensor nodes (cf. Section III). This greatly increases the resilience against node capture in contrast to previous schemes [3]–[8]. The algorithm for polynomial pre-distribution is described in Table I.

To calculate the size of F, we can view the set of all deployment points as a graph and each polynomial as an edge connecting two neighboring deployment points. For a size of $U \times V$ hexagonal grid, we can easily calculate the total number of edges is:

$$|F| = 3UV - 2U - 2V + 1 \ . \tag{5}$$

C. Link-Layer Key Establishment

After deployment, each sensor node broadcasts its node ID and the location (x, y) of the deployment point where it is supposed to reside. The broadcasted information can be in plaintext because adversaries know nothing about the associated polynomial even they overhear node IDs and the the location (x, y).

If two neighboring nodes find that they are destined to the same deployment point or two neighboring deployment points, they know that they belong to the same group or neighboring groups. Subsequently, they can establish a shared link-layer key by evaluating their own corresponding polynomial shares with the ID of each other as the input parameter. Since the ID of each node is unique, the established link-layer key is also unique. This property is particularly useful for secure communications in that it may not only provide perfect resilience to node compromise attacks, but also provide authentication service, so that the two nodes who have established the corresponding link-layer key may authenticate each other through the normal challenge-response method.

After the establishment of link-layer keys, the unused polynomial shares are removed to save memory resources, while the used polynomial shares are kept in nodes' memories. Such kept polynomial shares may be used to establish link-layer keys with new sensor nodes added in the future.

It is still possible that two neighboring nodes do not have shares of the same polynomial(s). In this case, they can rely on the secure multi-link paths between them to establish a link-layer 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 established a link-layer key. Because each link along the path is secure, it is safe to exchange a link-layer key between n_a and n_b with the help of the intermediate nodes. However, the link-layer key may be exposed if one of the nodes along the path is compromised. To deal with this situation, multi-path routing approaches such as *SPREAD* [10] can be applied to securely exchange the link-layer key between n_a and n_b. For the lack of space, The further investigation on this issue is left to the extension of this paper.

Apparently, our approach greatly deceases the broadcast overhead and thus saves the energy in contrast to previous key pre-distribution schemes [3]–[7], in which hundreds of or even thousands of key (space) IDs need to be broadcasted for two neighboring nodes to find a common key.

D. Network Extensibility

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 cancelled and their keys need to be revoked. We can easily achieve this by only removing the corresponding keys out of each node's memory.

We may need to add new nodes into the network in some cases and then we also need to establish link-layer keys for the new nodes. Recall that each node in the network is preloaded with 6 polynomials, but not all of them are used to establish link-layer keys immediately after deployment. Because the number of nodes in one group is no more than $\lfloor \frac{t}{2} \rfloor$, one polynomial can be used no more than t times. We can add new sensor nodes preloaded with the shares of the 6 polynomials used in the group that we want to deploy it to. After deployment, the new node can broadcast the location of its deployment point and its neighbors destined to the same deployment point or neighboring deployment points can establish link-layer keys with the new node. After that, the new node needs to delete the unused polynomials shares and keep the used polynomials shares. If a new node does not share polynomial shares of the same polynomials with some neighboring nodes, the new node can establish link-layer keys with these neighboring nodes by utilizing the aforementioned multi-path method.

If each polynomial is used no more than t times, the network is perfectly robust against node compromise. However, if we add too many new nodes, one of 6 polynomials may be kept by more than t nodes so that the polynomial may be exposed when adversaries compromise at least $t + 1$ nodes, each of which keeps one share of the polynomial. There are two methods to solve the problem. One method is to increase the degree of polynomials, thus we may decrease the possibility that the polynomials are exposed while increasing the extensibility of the network. The other method is to densely deploy sensor nodes with enough redundancy so that the necessity of node additions can be decreased.

VI. Analysis

Here we conduct some analysis of the proposed scheme in terms of network connectivity, memory cost, broadcast overhead, and network resilience against node compromise.

A. Network Connectivity

In sensor networks, a sensor node relies on its neighbors to relay messages, but cannot communicate directly with all its neighbors in a secure manner after deployment. We are interest in the network connectivity, which is the probability that two neighboring nodes can establish a link-layer key directly. For sensor networks, high network connectivity is preferred.

Suppose node $n^{uv} \in G_{uv}$ is located at (x, y). Let $A(n_j^{u'v'}, n^{uv})$ be the event that node $n_j^{u'v'} \in G_{u'v'}$ is a neighbor of n^{uv}, $B(n_j^{u'v'}, n^{uv})$ be the event that node $n_j^{u'v'}$ is a secure neighbor of n^{uv}, and $C(n_j^{u'v'}, n^{uv})$ be the event that node $n_j^{u'v'}$ is in the same group as n^{uv} or one of the neighboring groups of n^{uv}. By secure neighbors, we mean those neighboring nodes of one give node, say n^{uv}, that can directly establish link-layer keys with it.

The probability that $n_j^{u'v'} \in G_{u'v'}$ is a neighbor of node n^{uv} is the integral of the PDF $p^{u'v'}(x, y)$ over the circle around node n^{uv}, i.e.,

$$P(A(n_j^{u'v'}, n^{uv})) = \iint_{|n_j^{u'v'} - n^{uv}| \leq R} p^{u'v'}(x, y)\, dxdy \ ,$$

where R is the node transmission range which is the same for all the sensor nodes and $|n_j^{u'v'} - n^{uv}|$ denotes the distance between nodes $n_j^{u'v'}$ and n^{uv}.

Let $T_j^{u'v'}$ be the experiment:

$$T_j^{u'v'} = \begin{cases} 1 \quad , & A(n_j^{u'v'}, n^{uv}) \text{ happens;} \\ 0 \quad , & \text{otherwise.} \end{cases}$$

Then the average number of neighbors of node n^{uv} located at (x, y) is:

$$N^{uv}(x,y) = \sum_{n_j^{u'v'} \neq n^{uv}} E[T_j^{u'v'}] = \sum_{n_j^{u'v'} \neq n^{uv}} P(A(n_j^{u'v'}, n^{uv})),$$

where $E[T_j^{u'v'}]$ indicates the expectation of $T_j^{u'v'}$.

We can calculate the average number of secure neighbors of node n^{uv} located at (x, y) in the similar way, i.e.:

$$\begin{aligned} M^{uv}(x,y) &= \sum_{n_j^{u'v'} \neq n^{uv}} P(B(n_j^{u'v'}, n^{uv})) \\ &= \sum_{n_j^{u'v'} \neq n^{uv}} P(A(n_j^{u'v'}, n^{uv}) \bigcap C(n_j^{u'v'}, n^{uv})), \end{aligned}$$

Then the average number of neighbors of one node is:

$$\begin{aligned} N &= \sum_{u,v} P(n^{uv} \in G_{uv}) \int\int N^{uv}(x,y) p^{uv}(x,y)\, dxdy \\ &= \frac{1}{UV} \sum_{u,v} \int\int N^{uv}(x,y) p(x - x_u, y - y_v)\, dxdy, \end{aligned}$$

and the average number of secure neighbors of one node is:

$$\begin{aligned} M &= \sum_{u,v} P(n^{uv} \in G_{uv}) \int\int M^{uv}(x,y) p^{uv}(x,y)\, dxdy \\ &= \frac{1}{UV} \sum_{u,v} \int\int M^{uv}(x,y) p(x - x_u, y - y_v)\, dxdy. \end{aligned}$$

Hence, the network connectivity p can be calculated as

$$p = \frac{M}{N} \ . \qquad (6)$$

B. Memory Cost

Before deployment, each node is preloaded with 6 polynomial shares. Since one polynomial share is a t-degree univariate polynomial with $t+1$ coefficients, the total memory cost of each node is $6(t+1)$. However, the unused polynomial shares would be deleted after deployment, as a result of which the total memory cost is in fact less than $6(t + 1)$. If we want to save more memory, we can delete all the polynomial shares after link-layer keys are established. But in this case, it is not easy to add more nodes into the deployment area. One method to solve this problem is to densely deploy more sensor nodes so that we can increase the redundancy and decrease the necessity of node additions.

C. Broadcast Overhead

In wireless sensor networks, the constrained energy reservoir makes it necessary for each node to save energy. Since the energy consumption of each node is closely related to its message transmissions and receptions, we should try our best to decrease the messages needed to be transmitted and/or received by each node.

In schemes [3]–[7], hundreds of key IDs need to be broadcasted to establish pairwise keys after deployment. In our scheme, each sensor node only needs to broadcast the location (x, y) of the deployment point it is destined to and its own node ID. This greatly saves the energy compared with previous key pre-distribution schemes.

D. Resilience To Node Compromise

Our scheme uses t-degree bivariate polynomials to establish link-layer keys. If each polynomial is shared by no more than t sensor nodes, adversaries know nothing about the link-layer keys shared between non-compromised nodes, no mater how many nodes are compromised.

In our scheme, each polynomial is used by only two neighboring cells, so the number of nodes in one cell should not be larger than $\lfloor \frac{t}{2} \rfloor$ if we assume equal-size groups. Suppose the inter-cell distance (the distance between two deployment points) is D, the area of one group is $\frac{\sqrt{3}D^2}{2}$. Then the deployment density is,

$$\rho = \frac{G_c}{\frac{\sqrt{3}D^2}{2}} \leq \frac{2\lfloor \frac{t}{2} \rfloor}{\sqrt{3}D^2} \ . \qquad (7)$$

As long as the above inequality is satisfied, our scheme is perfectly resilient to node compromise.

In [8], perfect resilience to node compromise can also be achieved by using high degree polynomials. But it is not feasible due to the restrained memory resources of each node. Each polynomial is used in 5 cells in [8], while only in 2 cells in our scheme. As a result, if the deployment density is the same, the polynomial degree in [8] should be much higher than that in our scheme in order to provide the same degree of resilience against node compromise, i.e., the memory cost in [8] will be much higher than that of our scheme.

VII. Comparison With Related Work

In this section, we compare our scheme with Liu and Ning's scheme [8], which is the closest work to ours. The scheme proposed in [8] uses a square-grid-based deployment model, in which the whole deployment area is divided into many square cells. Compared to the hexagonal cell, each square cell can not achieve isotropic relationship with all its neighboring cells because of the different distances between two neighboring cells, thus leading to the lower network connectivity than our scheme. Fig. 2 shows the comparison of network connectivity versus the cell size which is normalized by the node's transmission range, where all nodes are deployed in a rectangular area($2000m \times 2000m$). the node deployment density is the same ($0.0025m^{-2}$) for both schemes, each

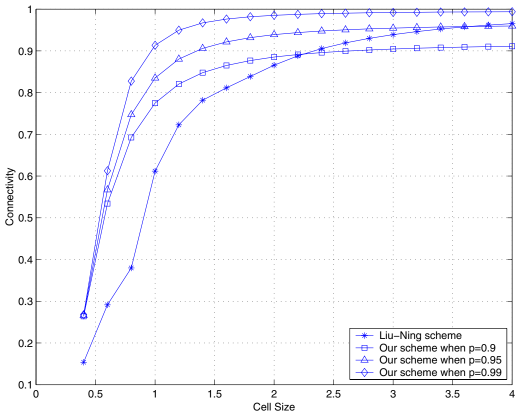

Fig. 2. Connectivity v.s. Cell size, where p is the probability that each node finally resides in its own cell area.

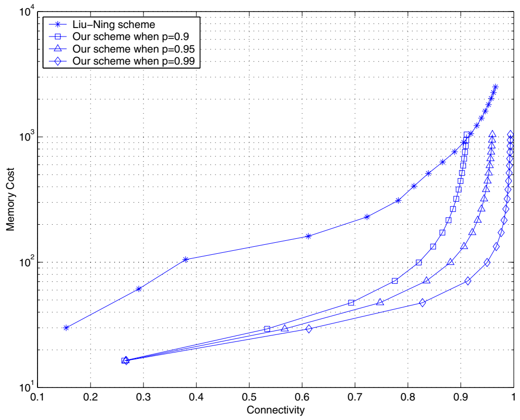

Fig. 3. Memory costs v.s. Connectivity, where perfect resilience to node compromise is guaranteed.

node in [8] is uniformly distributed in its entire cell(square shape), and each node in our scheme is distributed in its entire cell (hexagonal shape) with probability 0.9, 0.95 and 0.99, respectively. Apparently our scheme can achieve higher connectivity than [8] if the probability that each node resides in its own cell is properly set. We can achieve this by controlling the height of the aircraft which is used to deploy sensor nodes.

Next we compare the memory costs of the two schemes. In [8], each polynomial is used in 5 groups. Suppose the deployment density is ρ and the inter-cell distance is D for both schemes. The number of nodes keeping shares of one polynomial in [8] is $5\rho D^2$, while in our scheme the number is only $\sqrt{3}\rho D^2$ because each polynomial is used only in 2 neighboring cells. If we assume the same polynomial degree, [8] is more vulnerable to node compromise because each polynomial is used by more nodes. To get the perfect resilience to node compromise, the polynomial degree in [8] should be no less than $5\rho D^2$. Because each node holds shares of 5 polynomials, the memory cost of each node in [8] is no less than $5(5\rho D^2+1)$. However, in our scheme, the memory cost is only $6(\sqrt{3}\rho D^2+1)$. Even if [8] can tolerate the memory cost up to $5(5\rho D^2+1)$, from Fig. 2 we can see the connectivity of [8] may still be lower compared with our scheme when the inter-cell distance is same. To achieve the same connectivity, the inter-cell distance in [8] should be increased. This means the memory cost of [8] may be much higher. From Fig. 3, we can see that to achieve the same connectivity with perfect resilience to node compromise, our scheme has much less memory requirements than [8]. For example, when the deployment density is $0.0025m^{-2}$, to achieve the connectivity 0.95 with perfectly resilience to node capture, the memory cost of [8] is about 1734, while the memory cost of our scheme is about 433 and 100, with the possibility that each node resides in its own cell to be 0.95 and 0.99, respectively.

VIII. Conclusion And Future Work

In this paper we proposed an efficient scheme to establish link-layer keys between neighboring nodes, which are requisite for the secure communications in wireless sensor networks. Our scheme utilizes a hexagonal-grid-based deployment model for the first time to increase network connectivity, and uses bivariate polynomials locally to establish a link-layer key for each pair of neighboring nodes. In contrast to conventional schemes, our scheme can achieve higher network connectivity, increase network resilience to node compromise attacks, and decrease broadcast overhead with lower memory costs.

As the future research, we plan to further investigate on how to improve the network extensibility. We also intend to further study the influence of inter-cell distance and the node transmission range on the network connectivity. In addition, we will study how to implement other secure services based on our scheme.

References

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

[2] 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.

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

[4] 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.

[5] D. Liu, P. Ning, “Establishing pairwise keys in distributied sensor networks,” CCS’03 Washington, DC, 2003.

[6] 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.

[7] 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.

[8] D. Liu and P. Ning, “Location-based pairwise key establishments for relatively static sensor networks,” In ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN’03), October 2003.

[9] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung,“ Perfectly-secure key distribution for dynamic conferences,” In Advances in Cryptology C CRYPTO 92, LNCS 740, pages 471C486, 1993.

[10] Wenjing Lou, Wei Liu and Yuguang Fang, “SPREAD: Enhancing data confidentiality in mobile ad hoc networks,” IEEE INFOCOM 2004, HongKong, China, Mar 2004.

0-7803-8966-2/05/$20.00 (C) 2005 IEEE