SCALABLE LINK-LAYER KEY AGREEMENT IN SENSOR NETWORKS

Yun Zhou 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@, fang@ece.}ufl.edu

ABSTRACT

Link layer key agreement between neighboring nodes is a fundamental issue for securing sensor networks deployed in unattended and hostile environments. Recent research often assumed a pre-distributing approach so that each pair of neighboring nodes agrees on a shared key directly or indirectly through a multi-hop path. The shortcomings include large memory cost, poor resilience against node compromise, low secure connectivity, etc. In this paper, we present a link-layer key agreement scheme combining a scalable key agreement model and space diversity. In contrast to previous proposals, our scheme have higher resilience to node compromise attacks with much smaller memory costs and high secure connectivity so that much less energy needs to be consumed in establishing indirect keys through multi-hop routing.

INTRODUCTION

Key agreement is very critical for securing wireless sensor networks, because encryption and authentication services are based on the operations involving keys. The simple *pairwise key* approach, which requires each pair of nodes in a network with N nodes share a distinct symmetric key, is not scalable due to its memory cost of $N-1$ keys per node. In [1], [2], a threshold-based method is proposed such that each node stores $\lambda+1$ secrets and any pair of nodes is able to agree on a unique shared key. The collusion of less than $\lambda+1$ nodes can not reveal any key held by other normal nodes, i.e., λ-*secure*. To guarantee *perfect secure* in a network with N nodes, i.e., every key between a pair of nodes is secure no matter how many other nodes collude, the $(N-2)$-secure scheme should be used, which means the memory cost per node is $N-1$. Obviously, these schemes are lack of scalability and only suitable and optimum in small networks. Most recent research in sensor networks [3]–[14] focuses on the applications of those schemes in local area of sensor networks to establish *Link-layer keys* (called LLKs hereafter) among neighboring nodes, because the number of nodes in the neighborhood is limited.

In the pioneering work [3] and its followers [4]–[6], a global set of secrets is uniformly pre-distributed into the network so that each node has a secret subset and two neighboring nodes can achieve a *probabilistic* key agreement by the intersection of their secret subsets. In this way, two nodes can shared a key *directly* or establish an indirect key with the help of several intermediate nodes through a multi-hop path. These schemes are vulnerable to *node compromise* in that the secrets in compromised nodes can be used by the adversary to derive keys shared by other non-compromised nodes. Moreover, the number of secrets that each node needs to keep is limited due to the memory constraints, which implies a smaller probability that two neighboring nodes can establish a direct LLK, i.e., lower *local secure connectivity*, and much more communication overhead on indirect LLKs establishments through multi-hop paths.

Some *deterministic* work is developed in [7]–[9]. In [7], all N nodes in a network are organized into a 2 dimensional grid. Each node is preloaded with unique pairwise keys for $2(\sqrt{N}-1)$ nodes, which have the same horizontal or vertical coordinates. Any two neighboring nodes can establish an LLK with the help of no more than one intermediate node. Due to the uniform key pre-distribution, the local secure connectivity is very low, indicating the large energy consumption on the LLK establishment. Combinatorial design techniques are proposed in [8], [9]. They can ensure key sharing between any pair of nodes. In their schemes, however, each key is reused by many sensor nodes like that in [3]. This leads to poor resilience to node compromise. In addition, the memory cost of their schemes is roughly $\mathcal{O}(\sqrt{N})$ where N is the total number of nodes.

Some schemes [10]–[14] use location information to localize the impact of the node compromise attack and increase connectivity by intentionally pre-distribute the same set of secrets in small cells. They can achieve much higher connectivity than uniform pre-distribution schemes. However, they are probabilistic schemes, and still features vulnerability to node compromise and high memory cost.

In this paper, we propose a novel LLK agreement scheme that combines a scalable key agreement model we have developed in [15] and node deployment knowledge. Unlike the aforementioned schemes, our scheme is scalable in that it can can provide a high level of security and connectivity with very small memory cost, which is $\mathcal{O}(\sqrt[k]{N})$ per node, where $k \geq 1$.

The rest of the paper is organized as follows. We will describe

This work was supported in part by the US Office of Naval Research under grant N000140210464 (Young Investigator Award).

our scheme in Section II. Analysis and evaluation are carried out in Section III. Finally the paper is concluded in Section IV.

DETAILS OF OUR SCHEME

Our scheme is based on a method we have developed in [15]. Each sensor node carries a *share* of a global t-degree multivariate symmetric polynomial. If the shares of two nodes are correlated, the two nodes can calculate a shared key directly. Otherwise, they can negotiate an indirect key with the help of an intermediate node. We utilize *node deployment* knowledge such that nodes with correlated shares are deployed as close as possible. In this way, each node can directly calculate LLKs with most of its neighbors. In this section, we will elaborate the details of our scheme.

A. Mathematical Model

A t-degree $(k+1)$-variate polynomial is defined as

$$f(x_1, x_2, \ldots, x_k, x_{k+1}) = \sum_{i_1=0}^{t} \sum_{i_2=0}^{t} \cdots \sum_{i_k=0}^{t} \sum_{i_{k+1}=0}^{t} a_{i_1,i_2,\ldots,i_k,i_{k+1}} \, x_1^{i_1} x_2^{i_2} \cdots x_k^{i_k} x_{k+1}^{i_{k+1}} \, . \quad (1)$$

All coefficients of the polynomial are chosen from a finite field $\mathbb{F}_q$, where q is a prime that is large enough to accommodate a cryptographic key.

A $(k+1)$-tuple permutation is defined as a bijective mapping

$$\sigma : \quad [1, k+1] \longrightarrow [1, k+1] \, . \quad (2)$$

By choosing all the coefficients according to

$$a_{i_1,i_2,\ldots,i_k,i_{k+1}} = a_{i_{\sigma(1)},i_{\sigma(2)},\ldots,i_{\sigma(k)},i_{\sigma(k+1)}} \quad (3)$$

for any permutation σ, we can obtain a symmetric polynomial in that

$$f(x_1, x_2, \ldots, x_k, x_{k+1}) = f(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(k)}, x_{\sigma(k+1)}) \, . \quad (4)$$

Every node should have k credentials, which are *positive* and *pairwise different* integers. Suppose node u has credentials $(u_1, u_2, \ldots, u_k)$ and node v has credentials $(v_1, v_2, \ldots, v_k)$. Before node deployment, we can assign a *polynomial share* $f(u_1, u_2, \ldots, u_k, x_{k+1})$ to u and another share $f(v_1, v_2, \ldots, v_k, x_{k+1})$ to v. By assigning polynomial shares, we mean that the coefficients of t-degree univariate polynomials $f(u_1, u_2, \ldots, u_k, x_{k+1})$ and $f(v_1, v_2, \ldots, x_{k+1})$ are loaded into node u's and v's memory, respectively.

If the credentials of node u and node v have only one mismatch, i.e.,

1) for some $i \in [1, k]$, $u_i \neq v_i$, and
2) for $j = 1, 2, \ldots, k$, $j \neq i$, $u_j = v_j = c_j$,

then node u and node v can have a shared key. Node u can take v_i as the input to its share $f(u_1, u_2, \ldots, u_k, x_{k+1})$, and node v can take u_i as the input to its share $f(v_1, v_2, \ldots, v_k, x_{k+1})$. Due to the polynomial symmetry, the desired shared key between nodes u and v is calculated as

$$\begin{aligned} K_{uv} &= f(c_1, c_2, \ldots, c_{i-1}, u_i, c_{i+1}, \ldots, c_k, v_i) \\ &= f(c_1, c_2, \ldots, c_{i-1}, v_i, c_{i+1}, \ldots, c_k, u_i) \, . \end{aligned} \quad (5)$$

B. Network Model

We assume each node is identified by an index-tuple $(n_1, n_2, \ldots, n_k)$, where $n_i = 0, 1, \ldots, N_i - 1, i \in \{1, 2, \ldots, k\}$, and we may use the index-tuple as the *node ID*. Hence each node is mapped into a point in a k-dimension vector set $\mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_k$, where $n_i \in \mathcal{S}_i \subset \mathbb{Z}$ and the cardinality $|\mathcal{S}_i| = N_i$, for $i = 1, 2, \ldots, k$. The maximum number of nodes that the network can consist of is $N = \prod_{i=1}^{k} N_i$.

Due to the broadcast characteristics of radio communications, adversaries can easily eavesdrop any messages, either non-encrypted or encrypted, transmitted over the air between nodes. Adversaries may capture any node and compromise the secrets stored in the node. Furthermore, adversaries can use the compromised secrets to derive more secrets shared between other non-compromised nodes. We try to reduce the probability that the keys shared between non-compromised nodes are exposed when some nodes have already been compromised. To further evaluate the impact of node compromise, we assume the probability of the compromise of a node is p.

C. Share Distribution

Before network deployment, a global t-degree $(k+1)$-variate symmetric polynomial is constructed as is stated in Section II-A. This polynomial is used to derive shares for sensor nodes.

To achieve key agreement, every node n should have k credentials $(c_1, c_2, \ldots, c_k)$, which are positive and pairwise different as is required in Section II-A. These credentials can be created and preloaded into nodes before deployment. However, it requires additional memory space per node. Fortunately, the k credentials can be derived from the k indices in node ID $(n_1, n_2, \ldots, n_k)$ by a bijection, i.e.,

$$\begin{cases} c_1 &= n_1 + 1 \\ c_2 &= n_2 + 1 + N_1 \\ c_3 &= n_3 + 1 + N_1 + N_2 \\ & \vdots \\ c_{k-1} &= n_{k-1} + 1 + N_1 + \cdots + N_{k-2} \\ c_k &= n_k + 1 + N_1 + \cdots + N_{k-1} \end{cases} , \quad (6)$$

where $n_i = 0, 1, \ldots, N_i - 1$ for $i = 1, 2, \ldots, k$. Thus, the k credentials are drawn from different zones in that $c_1 \in [1, N_1]$ and $c_i \in [N_1 + \cdots + N_{i-1} + 1, \; N_1 + \cdots + N_i]$ for $i = 2, \ldots k$, which guarantee they are positive and pairwise different (Fig. 1).

For a node $(n_1, n_2, \ldots, n_k)$, a *polynomial share*

$$f_{k+1}(x_{k+1}) = f(c_1, c_2, \ldots, c_k, x_{k+1}) = \sum_{i_{k+1}=0}^{t} b_{i_{k+1}} \, x_{k+1}^{i_{k+1}} \quad (7)$$

Figure 1. Construction of positive and pairwise different credentials according to the equation (6). c_i are credentials derived from node ID.

is calculated, where

$$b_{i_{k+1}} = \sum_{i_1=0}^{t} \sum_{i_2=0}^{t} \cdots \sum_{i_k=0}^{t} a_{i_1,i_2,\ldots,i_k,i_{k+1}} \, c_1^{i_1} c_2^{i_2} \cdots c_k^{i_k} \qquad (8)$$

and $(c_1, c_2, \ldots, c_k)$ is mapped from $(n_1, n_2, \ldots, n_k)$ according to the equations (6). Then the polynomial share is assigned to the node. Here, the node only knows the $t+1$ coefficients of the univariate polynomial share, but not the coefficients of the original $(k+1)$-variate polynomial. Therefore, even if the marginal bivariate polynomial is exposed, the global polynomial is still safe if the degree t is chosen properly.

D. Node Deployment

According to Section II-A, two nodes can calculate a shared key if their credentials have only one mismatch in them. Due to the one-to-one mapping in the equations (6), two nodes u with ID $(u_1, u_2, \ldots, u_k)$ and v with ID $(v_1, v_2, \ldots, v_k)$ can directly calculate a shared key without any interaction if their IDs have only one mismatch. If the two nodes are within the radio coverage of each, then the key can be used as an LLK. Therefore, we need a deployment method that intentionally make nodes with only one mismatch in their IDs be deployed as close as possible. In such a way, each node can establish Link-layer keys with most of its neighbors.

Because node ID is an index-tuple $(n_1, n_2, \ldots, n_k)$, where $n_i = 0, 1, \ldots, N_i - 1, i \in \{1, 2, \ldots, k\}$, the network is logically constructed with k levels. The i-th level consists of $N_1 \times N_2 \times \cdots \times N_i$ cells, each of which has N_{i+1} subcells, i.e., $N_{i+1} \times \cdots \times N_k$ nodes, where $i = 1, 2, \ldots, k-2$. The $(k-1)$-th level consists of $N_1 \times N_2 \times \cdots \times N_{k-1}$ cells, each of which has N_k nodes. Here, the notation $(n_1, n_2, \ldots, n_i)$ can be seen as *cell ID* at the i-th level for $i = 1, 2, \ldots, k-1$. An example is illustrated in Fig. 2 (a), where $N1 = N2 = N3 = 9$.

To facilitate key agreement, the logical network topology is transformed into a real one, which decides the real node deployment model. Suppose a level-$(i-1)$ cell has $N_i = R_i \times C_i$ subcells. To do the transformation, the following two steps are taken:

1) in the first step, flip the even rows of the level-$(i-1)$ cell vertically;
2) in the second step, flip the even columns of the level-$(i-1)$ cell horizontally.

An example is depicted in Fig. 3. A cell at the $(i-2)$-th level has $N_{i-1} = 3 \times 5$ level-$(i-1)$ subcells (Fig. 3 (a)), each of which again has N_i subcells. By the two-step transformation, we get the real cell topology illustrated in Fig. 3 (b).

The entire network topology is constructed based on the two-step transformation. In this way, the network is divided into $N_1 \times N_2 \times \cdots \times N_{k-1}$ cells, where cells are located according to the space order determined by the two-step transformation. All the nodes are deployed into corresponding cells based on their IDs. The real network topology of the example in Fig. 2 (a) is illustrated in Fig. 2 (b).

E. Link-layer Key Agreement

As is stated before, two nodes u with ID $(u_1, u_2, \ldots, u_k)$ and v with ID $(v_1, v_2, \ldots, v_k)$ can directly calculate a shared key without any interaction if there is only one mismatch, say the i-th indices, in their IDs. Then node u can take $v_i + 1 + N_1 + \cdots + N_{i-1}$ as the input to its own share $f(c_1, c_2, \ldots, c_k, x_{k+1})$, and node v can as well take $u_i + 1 + N_1 + \cdots + N_{i-1}$ as the input to its share $f(c_1, c_2, \ldots, c_k, x_{k+1})$. The direct shared key between nodes u and v is then calculated as

$$\begin{aligned} K_{uv} &= f(c_1, \ldots, u_i + 1 + N_1 + \cdots + N_{i-1}, \\ &\qquad \ldots, c_k, v_i + 1 + N_1 + \cdots + N_{i-1}) \\ &= f(c_1, \ldots, v_i + 1 + N_1 + \cdots + N_{i-1}, \\ &\qquad \ldots, c_k, u_i + 1 + N_1 + \cdots + N_{i-1}) \,. \end{aligned} \qquad (9)$$

Because all node credentials of u and v are drawn from different subsets where any two subsets have no intersection and $u_i \neq v_i$, the $k+1$ credentials used to calculate the shared key are pairwise different, and the set of credentials is unique. Therefore the shared key calculated by the nodes u and v is unique, i.e., other nodes do not know the shared key.

Consider our deployment model. At the lowest level, the network is divided into $N_1 \times N_2 \times \cdots \times N_{k-1}$ cells. All the nodes in each of those cells have common ID prefix, which is the cell ID, and their node IDs are only different at the k-th position. Therefore, any pair of nodes in one cell can calculate a direct shared key. For example, two nodes (041) and (044) in cell (04) (Fig. 2 (b)) can calculate a shared key directly.

For two neighboring cells, if their cell IDs has only one mismatch, each node in one cell can find another node in the other cell such that the two nodes have only one mismatch in their node IDs, i.e., the two nodes can calculate a shared key directly. In the example Fig. 2 (b), node (041) in cell (04) can calculate a shared key directly with node (081) in cell (08). With the help of node (081), node (041) can indirectly establish a shared key with every other node in cell (08).

For two neighboring cells with two mismatches in their cell IDs, they are in the diagonal direction of each other and have a neighboring cell in common, which has only one mismatch in cell ID with each of them. In Fig. 2 (b), node (081) in cell (08) can indirectly negotiate a shared key with node (151) in cell (15) through node (181) in cell (18), because node (181) has direct keys with node (081) and (151) respectively. Then node (081) can indirectly negotiate a shared key with each of other nodes in cell (15) through node (151).

In our deployment model, each node can calculate direct LLKs with most of its neighbors because most of its neighbors are in

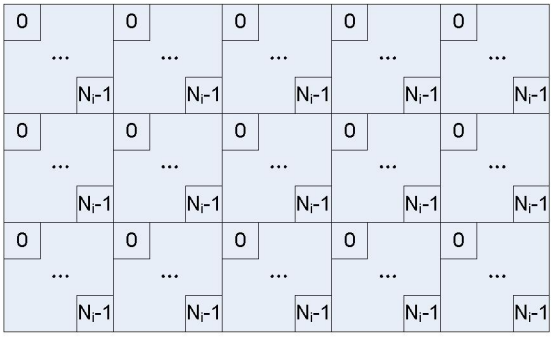
00	01	02	10	11	12	20	21	22
03	04	05	13	14	15	23	24	25
06	07	08	16	17	18	26	27	28
30	31	32	40	41	42	50	51	52
33	34	35	43	44	45	53	54	55
36	37	38	46	47	48	56	57	58
60	61	62	70	71	72	80	81	82
63	64	65	73	74	75	83	84	85
66	67	68	76	77	78	86	87	88

(a)

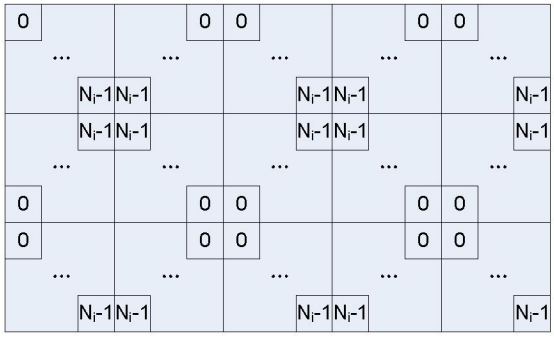
00	01	02	12	11	10	20	21	22
03	04	05	15	14	13	23	24	25
06	07	08	18	17	16	26	27	28
36	37	38	48	47	46	56	57	58
33	34	35	45	44	43	53	54	55
30	31	32	42	41	40	50	51	52
60	61	62	72	71	70	80	81	82
63	64	65	75	74	73	83	84	85
66	67	68	78	77	76	86	87	88

(b)

Figure 2. (a) Logical network topology before deployment. Each node has an ID (n_1, n_2, n_3), where $n_i \in [1, N_i]$ and $N_1 = N_2 = N_3 = 9$. (b) Real network topology after deployment. Each node (n_1, n_2, n_3) is deployed into the corresponding cell (n_1, n_2).



(a)



(b)

Figure 3. A cell at the $(i-2)$-th level has 15 level-i subcells, each of which again has N_i subcells. (a) is the logical topology before deployment. After flipping the even rows of the level-$(i-2)$ cell vertically and then the even columns horizontally, we get (b), the real topology for node deployment.

the same cell as it, and negotiate indirect LLKs with the rest neighbors with the help of only one intermediate node. Moreover, each node can even establish shared keys with other nodes multi-hop away.

Analysis And Evaluation

In this section, we will carry out some analysis and evaluate our scheme in comparison with some typical schemes including [3], [7]–[10].

A. Memory Cost

All nodes in the network hold partial information of one t-degree $(k+1)$-variate polynomial to achieve key agreement. The memory cost of each node is $t+1$ because each node needs to store a t-degree univariate polynomial. The smaller t is, the less memory cost. However, the value of t should not be too small. During the network lifetime, some nodes may be compromised and then collaborate to expose the polynomial with the partial information they hold whereby to directly calculate keys between other nodes. Obviously, the polynomial degree t is an indication of the difficulty to expose the polynomial, and it is directly related to the security performance. By choosing the value of t properly, we can guarantee that no matter how many nodes are compromised, their collaboration cannot expose direct keys held between other non-compromised nodes, i.e., the global t-degree $(k+1)$-variate polynomial cannot be exposed. It has been proved in [15] that to guarantee the security of the global polynomial the following two conditions must be satisfied:

$$0 \le N_i - 2 \le t\,, \qquad i = 1, 2, \ldots, k\,, \tag{10}$$

and

$$\frac{1}{2}\Big(\prod_{i=1}^{k} N_i\Big)\Big(\sum_{i=1}^{k} N_i + k\Big) \le \binom{t+k+1}{k+1}\,. \tag{11}$$

Obviously, the solution should be

$$t \ge t^*\,. \tag{12}$$

However, it is very difficult to get t^* analytically. If we let $N_i = N_1$ for $i = 1, 2, \ldots, k$, i.e., all subspaces have the same number of indices, we can bound t^* as [15]

$$t^* \le r \cdot N_1\,, \tag{13}$$

TABLE I. Bound and precise ratios between t^* and N_1

k	r	t^*/N_1
1	1	1
2	1.8171	1.7715
3	2.4495	2.3919
4	2.9926	2.9219
5	3.4878	3.4058

TABLE II. Memory cost of different schemes

Schemes	Memory Cost
E-G [3]	m
LBKP [10]	$5(t+1)$
PIKE [7]	$2(\sqrt{N}-1)$
Combinatorial schemes [8], [9]	$\mathcal{O}(\sqrt{N})$
Ours	$\mathcal{O}(\sqrt[k]{N})$

where ratio

$$r = \sqrt[k+1]{\frac{k(k+1)!}{2}} . \tag{14}$$

The second column in Table I gives some bound ratios when k is small. The numerical value of t^* are given in the third column in Table I when k is small.

We compare the memory cost per node of our schemes with other schemes in TABLE II. In E-G scheme [3] each node has a subset of m keys, where m may be more than 100 if it needs to maintain a certain security or connectivity. In LBKP [10] each node is preloaded with 5 polynomial shares, each of which has a degree of t. However, in order to maintain strong security, the value of t is very high. So its memory cost is much higher than ours. In PIKE [7], each node must store $2(\sqrt{N}-1)$ keys where N is the network size. Combinatorial design techniques are proposed in [8], [9]. They are similar to E-G [3], but they can ensure key sharing between any pair of nodes. The memory cost of their schemes is roughly $\mathcal{O}(\sqrt{N})$ where N is the total number of nodes. However, the memory cost of our scheme can be $\mathcal{O}(\sqrt[k]{N})$, which is much less.

B. Security

In our scheme, each node can calculate direct LLKs with most of its neighbors. Each direct LLK is only known by the pair of nodes that shares it, and the key can not be derived by other nodes, because we choose the value of t such that the global polynomial is secure in case of node compromise. As for other neighbors, each node can negotiate an indirect LLK with each of them through only one intermediate node. So if the probability of node compromise is p, then the probability of the exposure of the indirect key is just p.

In conventional schemes [3], [8]–[10], when the number of compromised nodes is large, the direct keys among non-compromised nodes can be exposed. Moreover, the indirect keys are as well insecure because each indirect key has to be established with the help of several intermediate nodes along a path. If such a path involve h intermediate nodes, then the probability that an indirect key is exposed can be calculated as

$$P_c = 1 - (1-p)^h . \tag{15}$$

PIKE [7] is similar to our scheme in that any pair of nodes can establish a shared key through no more than one intermediate node. The difference is that it does not utilize deployment knowledge to facilitate LLK agreement and thus is more expensive in communication. Moreover, its memory cost per node is higher than ours. Obviously, our scheme is more secure than conventional schemes.

C. Local Secure Connectivity

Every node can calculate direct LLKs with some neighbors, and establish indirect LLKs with other neighbors through one intermediate node. The local secure connectivity is directly related to the communication overhead of key establishments. If a node has high probability to calculate direct LLKs, it can save a lot of communication overhead on the establishment of indirect LLKs through multi-hop routing. Hence, high local secure connectivity, which is the probability of establishment of direct LLKs, is desirable in sensor networks.

Suppose nodes are uniformly deployed in each cell. The local secure connectivity can be calculated as the ratio of node coverage in its cell to the node transmission area. Suppose the side length of each cell is $2D$, node radio radius is R. Due to the symmetry of square cell, we only consider the first quadrant in the Cartesian coordinate plane (Fig. 4), where the center of cell is located at the origin of the plane. The first quadrant is divided into five areas, each of which is corresponding to different node coverage $A(x_o, y_o)$ in the cell, where (x_o, y_o) is the location of node. The $A(x_o, y_o)$ can be calculated as

$$A(x_o, y_o) = \tag{16}$$

$$\begin{cases} \pi R^2 , & \text{when } 0 \le x_o < D-R, 0 \le y_o < D-R \\ R^2(\pi - \frac{1}{2}\arccos(2{Y_o}^2 - 1) + Y_o\sqrt{1-{Y_o}^2}) , & \text{when } 0 \le x_o < D-R, D-R \le y_o < D \\ R^2(\pi - \frac{1}{2}\arccos(2{X_o}^2 - 1) + X_o\sqrt{1-{X_o}^2}) , & \text{when } D-R \le x_o < D, 0 \le y_o < D-R \\ R^2(\pi - \frac{1}{2}\arccos(2{X_o}^2 - 1) - \frac{1}{2}\arccos(2{Y_o}^2 - 1) & \\ +X_o\sqrt{1-{X_o}^2} + Y_o\sqrt{1-{Y_o}^2}) , & \text{when } D-R \le x_o < D, D-R \le y_o < D, \\ & (x_o-D)^2 + (y_o-D)^2 > R^2 \\ R^2((X_o + \sqrt{1-{Y_o}^2})(Y_o + \sqrt{1-{X_o}^2}) & \\ + \arccos(-X_o\sqrt{1-{Y_o}^2} - Y_o\sqrt{1-{X_o}^2}) & \\ -|\sqrt{(1-{X_o}^2)(1-{Y_o}^2)} - X_oY_o|) , & \text{when } D-R \le x_o < D, D-R \le y_o < D, \\ & (x_o-D)^2 + (y_o-D)^2 \le R^2 \end{cases} \tag{17}$$

where $X_o = \frac{D-x_o}{R}, Y_o = \frac{D-y_o}{R}$. Thus the local secure connectivity can be calculated as

$$C = \frac{1}{\pi R^2 D^2} \int_0^D \int_0^D A(x_o, y_o)\, dx_o dy_o . \tag{18}$$

In scheme [3], each node selects M keys from S keys, thus the local secure connectivity is roughly $1 - \binom{S-M}{M} / \binom{S}{M} \approx \frac{M^2}{S}$, where $S \gg M$. In PIKE [7], each node keeps unique pairwise keys with $2(\sqrt{N}-1)$ nodes, thus the local secure connectivity of PIKE is about $2/\sqrt{N}$. Schemes [8], [9] are similar to PIKE

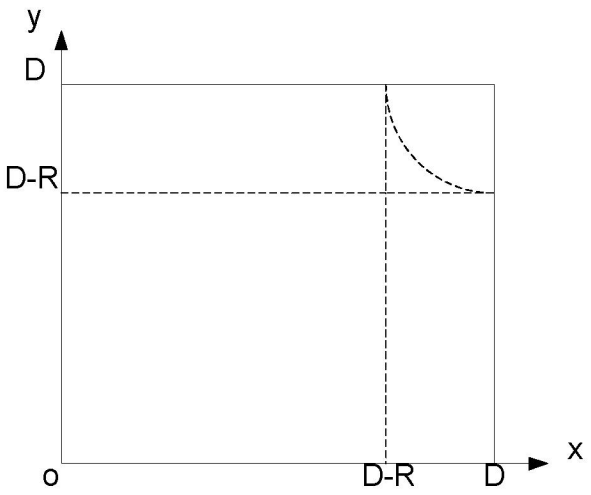


Figure 4. Node coverage in one cell.

[7] in that the local secure connectivity is roughly $\mathcal{O}(1/\sqrt{N})$. LBKP scheme [10] uses location information to facilitate key pre-distribution so that each node can establish direct LLKs with all neighbors in its cell and in neighboring cells, leading the local secure connectivity to about 1 if all the nodes are uniformly deployed in their home cell.

The low local secure connectivity of schemes [3], [7]–[9] is because each node cannot store too much keys to increase the local secure connectivity. However, in ours scheme the local secure connectivity is unrelated to memory cost. For example, suppose the size of cell size is $200 \times 200m^2$ and node radio radius is $25m$. The local secure connectivity of ours scheme is 0.89, which is much higher than that of [3], [7]–[9], which is usually much less than 0.5.

CONCLUSION

In this paper, we proposed a novel key agreement scheme, which is scalable for large networks with small memory cost. Compared with conventional schemes which have memory cost of at least $\mathcal{O}(\sqrt{N})$ in a network with N nodes, our scheme has only $\mathcal{O}(\sqrt[k]{N})$ memory cost per node, where $k > 1$. Moreover, we utilize node deployment knowledge to facilitate direct LLK agreement so that the local secure connectivity is very high. In this way, the communication overhead of establishing indirect LLKs is reduced significantly. The security of our scheme is very strong in that most LLKs are established directly, and the other indirect LLKs are established through only one intermediate node.

REFERENCES

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

[2] 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 471-486, 1992.

[3] L. Eschenauer and V. Gligor, "A key management scheme for distributed sensor networks," in *ACM CCS'02*, Washington D.C., 2002.

[4] H. Chan, A. Perrig, and D. 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] W. Du, J. Deng, Y. S. Han, and P. K.Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," in *CCS'03*, Washington, DC, October 27-30, 2003.

[6] D. Liu, and P. Ning, "Establishing pairwise keys in distributied sensor networks," *CCS'03*, Washington, DC, 2003.

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

[8] J. Lee and D. Stinson, "Deterministic key predistribution schemes for distributed sensor networks," *Selected Areas in Cryptography*, 2004.

[9] S. Camtepe and B. Yener, "Combinatorial design of key distribution mechanisms for wireless sensor networks," *Proc. Ninth European Symp. Research Computer Security*, 2004.

[10] 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.

[11] 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.

[12] Y. Zhou, Y. Zhang, and Y. Fang, "LLK: A link-layer key establishment scheme in wireless sensor networks," in *IEEE WCNC'05*, New Orleans, LA, March 2005.

[13] Y. Zhou, Y. Zhang, and Y. Fang, "Key establishment in sensor networks based on triangle grid deployment model," *IEEE Military Communications Conference (MILCOM'05)*, Atlantic City, New Jersey, October 17-20, 2005.

[14] D. Liu, P. Ning and W. Du, "Group-based key pre-distribution in wireless sensor networks," *ACM WiSe'05*, September, 2005.

[15] Y. Zhou and Y. Fang, "A scalable key agreement scheme for large scale networks," in *2006 IEEE International Conference on Networking, Sensing and Control* (ICNSC06), Ft. Lauderdale, Florida, USA, April 23-25, 2006.