MKPS: A Multivariate Polynomial Scheme for Symmetric Key-Establishment in Distributed Sensor Networks

Farshid Delgosha
Dep. of Electrical and Computer Eng.
Georgia Institute of Technology
Atlanta, GA 30332–0250
Email: delgosha@ieee.org

Erman Ayday
Dep. of Electrical and Computer Eng.
Georgia Institute of Technology
Atlanta, GA 30332–0250
Email: erman@ece.gatech.edu

Faramarz Fekri
Dep. of Electrical and Computer Eng.
Georgia Institute of Technology
Atlanta, GA 30332–0250
Email: faramarz.fekri@ece.gatech.edu

***Abstract*—Privacy is a critical service in node-to-node communications when sensor networks are deployed in adversarial environments. However, providing this service is a nontrivial task because of the lack of infrastructure and node limitations. Existing techniques distribute secret keys to the network users through a trusted third party or using computationally-complex public-key methods. An alternative approach is pre-distributing keying material to the nodes prior to the network deployment. Exploiting the mathematical properties of symmetric polynomials, we propose a multivariate key pre-distribution scheme (MKPS) in this paper. In this scheme, using uniquely assigned IDs, shares of d-variate polynomials are stored into the memory of every sensor. After the network deployment, every two neighbor nodes at the unit Hamming distance of each other establish exactly $d-1$ common keys without any interaction with a third party in the network. The final secret key used by these nodes is a symmetric combination of all the common keys. We will show that this feature significantly improves the security of the MKPS over previous schemes. The proposed method is in the category of threshold schemes, i.e., it remains perfectly secure up to the capture of a certain fraction of sensor nodes. We also propose a location-aware MKPS in which, by taking advantage of the location information, perfect connectivity is achieved. The new location-aware scheme is a cell-based method in which nodes are randomly deployed within hexagonal cells. Nodes are unaware of their exact locations. Nevertheless, they know the coordinates of their residing cells. One MKPS is used to secure communications within every cell and one to secure communications between cells. This location-based scheme significantly improves the resiliency of the network against the node capture.**

I. Introduction

In the era of information technology and with the advent of micro-electro-mechanical systems and low power highly integrated electronic devices, wireless sensor networks (WSNs) are expected to play key roles in many applications such as managing energy plants, logistics and inventory, battlefields, and medical monitoring [1]. A typical sensor network may consist of hundreds to several thousands of sensor nodes that are low cost and battery powered, and have limited computation power and memory. Sensor nodes are either randomly or manually scattered in a field. They form an unattended wireless network that collects information about the field such as temperature, illumination, motion, some chemical material, etc. The collected data is partially aggregated and forwarded to a central processing unit, called the sink, that is responsible for interpreting the data and taking appropriate actions (e.g., sending personnel for precise measurements).

In hostile environments, security services are critical for a WSN to function healthy. However, the security of such networks poses new challenges because of the sensor constraints and networking features. Neighboring nodes in the sensor networks often experience correlated events. Thus, to conserve the transmission energy, sensor networks require in-network processing, aggregation, and duplication elimination. This imposes the need for a trusted connection between neighboring nodes, which are not considered in ad-hoc networks.

Since sensor nodes are power- and memory-starved devices, only a fraction of memory and computational power can be devoted to cryptographic algorithms to provide secure communication. Because of the computational cost of public-key systems, symmetric-key schemes become the tools of choice in sensor networks to provide data confidentiality and authentication at the link layer. A fundamental open research problem is to set up (pairwise) secret keys among the communicating nodes, referred to as *key establishment*, that is required for symmetric cryptographic schemes.

A classical solution is using a key-distribution center (KDC) to distribute the secret keys. A KDC, if employed in a network, has to directly communicate with all users. However, the massive number of sensors deployed in a field would introduce so much traffic at the KDC. Moreover, the short communication range of sensor nodes and also their widespread distribution makes direct communication with the KDC impossible. Hence, this solution is infeasible in WSNs.

Another solution is pre-distributing the keys among the sensor nodes in the network; hence, so called key pre-distribution schemes (KPSs). Although this solution seems feasible at the first glance, its realization is not trivial because of the massive number of nodes in the network and their resource limitations. A naive approach is using a single key to secure the communication traffic in the entire network. Nevertheless, this approach must be avoided since capturing only one node compromises the entire network to the adversary. Another approach, in a network of size n, is storing $n-1$ pairwise keys in every node, each for communicating with one other node in the network. However, in a typical sensor network with thousands of sensor nodes, the storage requirement for this approach is beyond the memory limitations of sensors. Moreover, not every two sensors are in the communication range of each other. If the neighbor sensors of every sensor in the field are known prior to the deployment, then it suffices to store only pairwise keys for the communication with neighbor sensors. However, such information are unavailable in typical sensor networks.

A. Related Work

The first practical KPS for sensor networks is the Eschenauer–Gligor (EG) scheme [2]. In this scheme, a large pool of keys is generated at the server prior to the network deployment. For every sensor node, a small fraction of keys, called the key ring, is randomly selected from the key pool and is stored in the sensor memory. Every two sensor nodes that happen to have at least one common key in their key rings are able to establish a secure communication link. In order to improve the resiliency of this scheme against the node

This material is based upon work supported by the Army Research Office under Grant No. 49586CI

capture, many modifications have been suggested in [3]–[7]. One of the modifications to the EG scheme is the q-composite scheme of [3]. In this scheme, every two nodes are enforced to have at least q common keys in their key rings to establish a secure link. The final common secret key is a symmetric combination of the q common keys.

The application of threshold cryptography in the KPSs for general networks was first proposed in [8] and further studied in [9]. In the simplest form of such schemes, every sensor stores a share of a symmetric bivariate polynomial. The symmetry property of polynomials allows obtaining the same key by two sensors that have shares of the same bivariate polynomial. The adversary, who does not know the bivariate polynomials, has to capture at least a certain number of sensors to reconstruct a bivariate polynomial from its shares. A random KPS is proposed in [6] based on this idea.

One problem with the random KPS is that they do not guarantee key establishment between any two nodes even if their communication ranges are assumed unlimited. To solve this problem, a deterministic hypercube-based scheme (HBS) is proposed in [6]. This scheme improves the network connectivity by uniquely assigning points on a hypercube to all the sensor nodes as their IDs, which are used to distribute shares of multivariate polynomials. When the communication ranges of all sensor nodes are assumed unlimited, this KPS guarantees the establishment of a link between any two nodes.

B. Outline of Our Scheme

In this paper, we propose a multivariate key pre-distribution scheme (MKPS). In this scheme, a large set of symmetric multivariate polynomials is generated by the sink prior to the network deployment. Every sensor node is uniquely assigned an ID that is a d tuple consisting of nonnegative integers. These IDs are used to assign d d-variate polynomials to every node. For every node, the shares of these polynomials are stored in its memory. We will show that in this setting every two nodes with IDs at the Hamming distance of one from each other have shares of the same $d-1$ multivariate polynomials. Using these shares, these nodes can establish $d-1$ common keys. We note that this feature is obtained for free without any payoffs such as additional memory. The final secret key between these two nodes, called a link key, is a symmetric combination of all these $d-1$ common keys. This feature significantly adds to the security of the proposed KPS since an adversary has to compromise all the $d-1$ common keys in order to compromise a link key.

Taking advantage of the deployment knowledge, we propose a location-aware KPS that provides perfect connectivity. In this scheme, the entire terrain is divided into non-overlapping hexagonal cells. Two layers of MKPS provide connectivity to the network for the inner and intra-cell communications. The resiliency of this scheme against the node capture is better than the previously proposed schemes.

In brief, the contributions of this paper are:

1) We propose a novel KPS for sensor networks using multivariate polynomials that significantly increase the security of the scheme without increasing the size of the required node memory. Moreover, since every node in this scheme is assigned a unique ID, node-to-node authentication is obtained for free. The proposed scheme is scalable, i.e., the addition of new nodes to the network after its deployment is possible.
2) The ID of every node is a d tuple of nonnegative integers. Every two nodes with IDs at the Hamming distance of one from each other can establish exactly $d-1$ common keys. The final secret key between these nodes, called a link key, is a symmetric combination of all the $d-1$ common keys. This interesting feature, gained by employing multivariate polynomials, is obtained for free. In addition, it considerably improves the security in our scheme since an adversary has to compromise all the $d-1$ common keys to compromise a link key.
3) Taking advantage of the deployment knowledge, where this information is available, we modify the proposed MKPS to a double-layered KPS that provides perfect connectivity to the network. The new location-aware scheme, similar to its random counterpart, has a threshold effect that significantly improves the resiliency of the network against the node capture.

C. Notation

For any $d \in \mathbb{N}$, we define $[d] := \{ x \in \mathbb{Z} : 0 \le x \le d-1 \}$. For any ordered tuple $I = (i_0, \ldots, i_{d-1})$ and an arbitrary subset $\mathcal{J} \subsetneq [d]$, we define the reduced ordered-tuple $I\langle \mathcal{J} \rangle := (i_j : j \in [d] \backslash \mathcal{J})$, which is an ordered tuple with coordinate indices in the set $\mathcal{J}$ removed. For simplicity, we use the short forms $I\langle j \rangle := I\langle \{ j \} \rangle$ and $I\langle j, \ell \rangle := I\langle \{ j, \ell \} \rangle$. For an arbitrary set $\mathcal{A}$, the Hamming distance between two d-tuples $I, I' \in \mathcal{A}^d$ is a mapping $d_h : \mathcal{A}^d \times \mathcal{A}^d \to \{ 0, 1, \ldots, d \}$ such that $d_h(I, I')$ is the number of coordinates in which I and I' are different. The Galois field with prime order p is denoted by $\mathbb{F}_p$.

II. Multivariate Key Pre-distribution

Prior to introducing the proposed KPS, we define a *d-conference t-secure* scheme as follows [9].

Definition 2.1: Let $\mathcal{U}$ be a set of n users, and $d, t \in \mathbb{N}$ such that $d \le n$. A KPS for $\mathcal{U}$ is d-conference t-secure if:

1) Every subset $\mathcal{V} \subseteq \mathcal{U}$ of d users can compute a group key by cooperating with each other (e.g., exchanging their IDs).
2) The coalition of every set $\mathcal{A} \subset \mathcal{U} \setminus \mathcal{V}$ of $|\mathcal{A}| \le t$ users reveals no information about the group key established by the d users in $\mathcal{V}$.

Consider a network with the user set $\mathcal{U} = \{ U_0, \ldots, U_{n-1} \}$ in which every user has an ID that is an integer in $[n]$. The server generates a symmetric polynomial $f(x_0, \ldots, x_{d-1})$ in $d \le n$ variables of degree t in each variable with coefficients from the finite field $\mathbb{F}_p$. The polynomial f is symmetric in the sense that $f(x_{\sigma(0)}, \ldots, x_{\sigma(d-1)}) = f(x_0, \ldots, x_{d-1})$ for any permutation σ on d elements. The server assigns the coefficients of the polynomial $f_i(x_1, \ldots, x_{d-1}) := f(i, x_1, \ldots, x_{d-1})$ to user U_i for all $i \in [n]$. Since each polynomial f_i has at most $\binom{t+d-1}{d-1}$ monomials, the maximum storage-memory requirement for each user is $\binom{t+d-1}{d-1} \log_2 p$ bits. This scheme is optimal in the sense that the amount of information stored in each user is minimal [9].

Any set of at least d users are able to establish a common key using the polynomials in their memories. To see how, consider the set of users $\{ U_i : i \in \mathcal{I} \}$, where $\mathcal{I} \subseteq [n]$ is an arbitrary subset of size d. The users in this set, first, exchange their IDs. Then, each user U_i evaluates its polynomial f_i at $I\langle i \rangle$. By the symmetry property of f, the group key is $k_{\mathcal{I}} = f(\mathcal{I})$.

Along the lines of this idea, we present the MKPS that consists of two main phases. The *setup* phase, performed by the sink before the network deployment, is the one in which the IDs of sensor nodes are assigned. The other phase, *link-key establishment*, is performed by the nodes after the network deployment.

A. Setup

Let n be the maximum number of sensor nodes in the network. The first task is to assign a unique ID to every sensor node. A node

ID is a d tuple $I = (i_0, \ldots, i_{d-1})$, where $i_0, \ldots, i_{d-1} \in [\, m \,]$ and $m := \lceil \sqrt[d]{n} \rceil$. The sink randomly generates dm symmetric d-variate polynomials $f_i^j(x_0, \ldots, x_{d-1}t) \in \mathbb{F}_p[x_0, \ldots, x_{d-1}]$, where $i \in [\, m \,]$ and $j \in [\, d \,]$, with degree t in each variable. For the node with ID $I = (i_0 \ldots, i_{d-1})$, the sink forms the set

$$\mathcal{P}_I := \left\{ f_{i_j}^j(I \langle j \rangle, x_{d-1}) \quad \in \mathbb{F}_p[x_{d-1}] : \forall j \in [\, d \,] \right\} \ . \tag{1}$$

The following example illustrates how the set $\mathcal{P}_I$ is formed.

Example 2.1: Let $d = 3$. For a node with ID $I = (i_0, i_1, i_2)$, the set $\mathcal{P}_I$ consists of the polynomials $f_{i_0}^0(i_1, i_2, x_2)$, $f_{i_1}^1(i_0, i_2, x_2)$, and $f_{i_2}^2(i_0, i_1, x_2)$.

The coefficients of all polynomials in $\mathcal{P}_I$ are stored in the node I along with its ID. Since $|\mathcal{P}_I| = d$, the required storage memory for every node is $d(t+1)\log_2 p + d \log_2 m$, which is the same as the memory requirement in the HBS [6]. We note that the sets of polynomials assigned to the nodes are deterministically selected in our scheme. Hence, from this view point, our scheme is deterministic comparing to the EG and q-composite schemes in which the key rings are randomly selected from the key pool.

B. Link-Key Establishment

Every two nodes at the Hamming distance of one from each other are able to establish a shared key. Consider two nodes I and I' that differ only at the j-th coordinate, i.e., $d_h(I, I') = 1$. These nodes can establish the following $d-1$ common keys.

$$\begin{aligned} k_{I,I',\ell} &= f_{i_\ell}^{\ell}(I \langle j, \ell \rangle, i_j, i'_j) \\ &= f_{i_\ell}^{\ell}(I' \langle j, \ell \rangle, i'_j, i_j), \quad \forall \ell \in [\, d \,] \setminus \{\, j \,\} \end{aligned} \tag{2}$$

The final secret key established between these two nodes, referred to as the *link key*, is

$$k_{I,I'} = \mu(k_{I,I',\ell} : \forall \ell \in [\, d \,] \setminus \{\, j \,\}) \ , \tag{3}$$

where $\mu(x_0, \ldots, x_{d-2}) := h(x_0 || \cdots || x_{d-2})$ and h is a pre-image resistant hash function. The following example shows how a link key is established.

Example 2.2: Let $d = 3$. Consider the two nodes $I = (i_0, i_1, i_2)$ and $I' = (i_0, i_1, i'_2)$. They can establish exactly two common keys $k_{I,I',0} = f_{i_0}^0(i_1, i_2, i'_2) = f_{i_0}^0(i_1, i'_2, i_2)$ and $k_{I,I',1} = f_{i_1}^1(i_0, i_2, i'_2) = f_{i_1}^1(i_0, i'_2, i_2)$. The link key is $k_{I,I'} = h(k_{I,I',0} || k_{I,I',1})$.

We note that using d-variate polynomials in our scheme has created the condition that every two nodes at the Hamming distance of one from each other can establish exactly $d-1$ common keys. Hence, our scheme is in some sense $(d-1)$ composite. As we will show later, this feature greatly lowers the probability of the link-key compromise. Fortunately, this feature is obtained for free without any payoffs. This can be compared to the original q-composite scheme obtained from the EG scheme by requiring that every two nodes share at least q keys in their key rings to establish a link key. Although this restriction decreases the probability of link-key compromise for small numbers of captured nodes, it has the opposite effect when the number of captured nodes increases. This is because the q-composite scheme is probabilistic, and to increase the probability of sharing q keys between any two nodes, the size of the key pool must shrink. Thus, capturing a large number of nodes compromises more links than capturing a small number of nodes. However, our scheme is deterministic, and sharing $d-1$ keys between any two nodes with Hamming distance of one from each other is guaranteed by the structure of our scheme.

III. Evaluation of the MKPS

In this section, we evaluate the proposed MKPS in terms of the network connectivity and the probability of the link-key compromise. Throughout the section, we assume that n is the actual number of the sensor nodes in the network and $m = \lceil \sqrt[d]{n} \rceil$.

A. Network Connectivity

The network is connected if there exists a link or a path connecting any two nodes. Since a path is a sequence of nodes that are consecutively connected with links, the probability of the network connectivity depends on the average probability of the link-key establishment denoted by P_{lk}. As explained before, in the MKPS, every two nodes at the Hamming distance of one from each other can establish a link key. Hence, the average probability of the link-key establishment is[1]

$$\begin{aligned} P_{lk} &\approx \frac{[d(m-2)+\nu]\,(m-1)^d}{n(n-1)} \\ &\quad + \frac{\theta m^{d+1}\left[d-2+\nu\theta^{\nu-1}+(2-d-\nu)\theta^{\nu}\right]}{n(n-1)} \\ &\leq \frac{d(m-1)}{n-1} \ , \end{aligned} \tag{4}$$

where

$$\theta := 1 - 1/m \qquad \nu := \left\lfloor \frac{\log\left(1+\theta^d - nm^{-d}\right)}{\log\theta} \right\rfloor \ . \tag{5}$$

This probability is plotted in Figure 1 versus the total number of nodes n and the dimensionality d. The abrupt changes in this figure are due to the ceiling function $\lceil \cdot \rceil$ in the definition of m. As the figure shows, for a fixed dimension d, the average probability of link-key establishment P_{lk} decreases by increasing the number of nodes n in the network. However, by fixing n and increasing d, the probability P_{lk} is globally decreasing, but it has linearly increasing segments. It globally decreases because m exponentially decreases with d although there is a linear coefficient of d in the first term of P_{lk} in (4). The linear increasing segments of P_{lk} correspond to the range of n for which m is constant and henceforth, the linear coefficient of d has the dominant effect.

Similar to the previous work [2], [3], [5], [6], we have used the random graph model $\mathcal{G}(n, P_{lk})$ in the derivation of (4). In this model, the communication radius of every node is assumed unlimited. However, in practice, every node has only a limited communication

[1]Because of the space limitation, details of the derivation of this equation are omitted here.

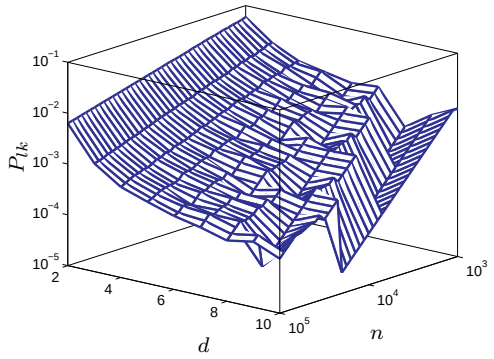

Fig. 1. Average probability of the link-key establishment

range. The relationship between the actual communication radius of the sensor nodes required for connectivity and the probability of the link-key establishment in the $\mathcal{G}(n, P_{lk})$ model is studied in [10]. As proved here, the minimum communication radius required to have a connected network is $R = \sqrt{(\ln n + \zeta) / (\pi n P_{lk})}$, where $\zeta > 0$ is a constant.

B. Resilience Against Node Capture

The adversary may attempt to disrupt the network connectivity by splitting the network into small components. This goal can be achieved by physically capturing some nodes in the network and deriving information about the secret keys used to secure communications between un-captured nodes. Adversary has to compromise all the $d-1$ common keys established between two arbitrary nodes to compromise the link key established between them. A common key is obtained by evaluating shares of multivariate polynomials at the node IDs. We note that these shares are stored only in the memories of the two nodes establishing the common keys. Thus, without capturing these nodes, the adversary has to recover some variables of the multivariate polynomials, generating these shares, by capturing other nodes in the network that store the shares. The parameters of the scheme determine the minimum number of variables that can be recovered and, consequently, the minimum number of nodes that must be captured. This produces a threshold effect, i.e., prior to capturing a least number of nodes, the adversary is unable to compromise any link keys. In the following, we determine the security threshold of the proposed MKPS. Using this threshold, we calculate the probability of the link-key compromise and compare it to other schemes.

Consider the two nodes I and I' that differ only in the j-th coordinate, i.e., $d_h(I, I') = 1$. These nodes can establish a link key. By (3), the link key $k_{I,I'}$ is a function of $d-1$ common keys $k_{I,I',\ell}$. Hence, the adversary has to compromise all these $d-1$ keys generated by the polynomials $f^{\ell}_{i_\ell}(I\langle\ell\rangle, x_{d-1})$ and $f^{\ell}_{i_\ell}(I'\langle\ell\rangle, x_{d-1})$ as in (2). However, these polynomials are stored only in the memories of I and I' that are unavailable to the adversary. Thus, for every $\ell \in [d]\backslash\{j\}$, the adversary has to recover the polynomial

$$f_\ell(x_{d-r-1}, \ldots, x_{d-1}) = f^{\ell}_{i_\ell}(\hat{\imath}_0, \ldots, \hat{\imath}_{d-r-2}, x_{d-r-1}, \ldots, x_{d-1}) \quad (6)$$

from its shares distributed in the network for some integer $1 \le r \le d-1$, where $(\hat{\imath}_0, \ldots, \hat{\imath}_{d-2}) = I\langle\ell\rangle$. The shares of this polynomial are $f_\ell(\hat{\imath}_{d-r-1}, \ldots, \hat{\imath}_{d-2}, x_{d-1})$, where $\hat{\imath}_{d-r-1}, \ldots, \hat{\imath}_{d-2} \in [m]$. These shares are accessible to the adversary upon capturing other sensor nodes. There are at most m^r shares of this polynomial available in the network. The minimum number of shares required to recover this polynomial is given by the following lemma.

Lemma 3.1: To recover the polynomial f_ℓ in (6) from its shares, the minimum number of required shares is

$$\lambda(r, t) = \binom{t+r}{r}, \qquad 1 \le r \le d-1. \quad (7)$$

Proof: We note that $f_\ell(x_{d-r-1}, \ldots, x_{d-1}) = \sum_{i=0}^{t} f_{\ell i}(x_{d-r-1}, \ldots, x_{d-2})\, x_{d-1}^i$, where each coefficient $f_{\ell i}(x_{d-r-1}, \ldots, x_{d-2})$ is an r-variate symmetric polynomial of degree t in each variable that has $\binom{t+r}{r}$ coefficients. Hence, $\lambda(r, t)$ shares are required. ∎

If $m^r < \lambda(r, t)$ for some r, then there are not enough shares of the polynomial in the network to recover r variables. Therefore, it is impossible for an adversary to obtain the polynomial in (6) for the given value of r. We note that the inequality $m^r < \lambda(r, t)$ does not imply $m^{r+1} < \lambda(r+1, t)$. In other words, we might have $m^{r+1} \ge \lambda(r+1, t)$ in which case there exist enough shares of f to recover $r+1$ variables. Hence, the security threshold is determined by the number of variables for which there exist enough shares in the network to recover that many variables. This result is summarized in the following corollary.

Corollary 3.1: Let $\mathcal{I} := \{ i \in \{1, 2, \ldots, d-1\} : m^i \ge \lambda(i, t) \}$ and $r := \min \mathcal{I}$. Then, the MKPS is $(\lambda(r, t) - 1)$-secure in the network.

The probability of compromising the link key established between any two nodes depends on the security threshold of the scheme. Assume that a fraction of p_{nc} nodes in the network is captured and r is given as in Corollary 3.1. By Lemma 3.1, the adversary has to obtain at least $\lambda(r, t)$ shares of any polynomial to recover r variables of that polynomial. Since the number of shares of a polynomial is a binomially-distributed random variable, the probability of polynomial recovery is

$$P_{pr} = \sum_{i=\lambda(r,t)}^{m^r} \binom{m^r}{i} p_{nc}^i \, (1 - p_{nc})^{m^r - i}. \quad (8)$$

To compromise a link key, all the $d-1$ common keys must be compromised. Hence, the probability of the link-key compromise is

$$P_{lkc} = P_{pr}^{d-1}. \quad (9)$$

The probability of link-key compromise versus the fraction of captured nodes is plotted in Figure 2 for different values of d. The degree t of the polynomials used in the scheme is adjusted with respect to the dimension d, by fixing the node memory to 30, to have a fair comparison. As the figure shows, by increasing the dimension, the scheme becomes more resistant against node capture. This is because the number of common keys constructing a link key increases.

Probabilities of the link-key compromise in the EG scheme of [2], the q-Composite of [3], the HBS of [6], and the proposed MKPS are compared to each other in Figure 3. For a fair comparison, in plotting all these curves, the node memory and the probability of the link-key establishment are fixed to 50 and $P_{lk} \approx 10^{-4}$, respectively. As this figure shows, the MKPS provides the highest resiliency against the node capture.

IV. Location Aware MKPS

There are many different ways to deploy a WSN in a field. For example, the deployment may be random in which there is

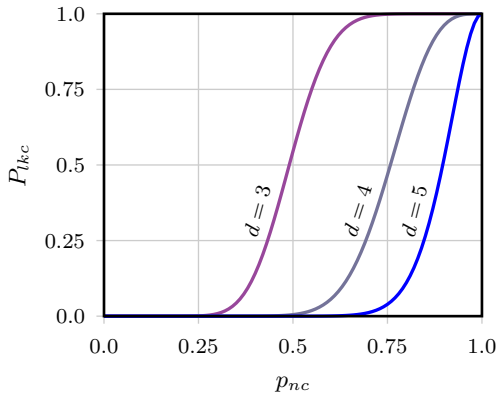

Fig. 2. Probability of the link-key compromise versus the fraction of captured nodes in the MKPS. Other parameters are $n = 10{,}000$ and $t = \lfloor 30/d - 1 \rfloor$.

no prior knowledge as to which sensors will be located at the vicinity of each other. In practice, usually a systematic deployment method is employed. For example, a group-based deployment model is suggested in [11]. In this model, the sensors are divided into groups of equal sizes. In addition, the field is covered with non-overlapping cellular areas. Sensors of each group are uniformly deployed in one cellular area. Using this method, the exact location of a sensor on the field is not known, but it is known which sensors are located in the same group. Hence, the deployment distribution is not uniform. It is possible to use this information toward KPS design to yield schemes that are more efficient than those designed without the deployment knowledge. A few location-aware schemes are proposed in the literature [11]–[13]. Using the MKPS scheme proposed in the previous section, we propose a location-aware KPS that is referred to as location-aware MKPS (LA–MKPS).

Usually sensors use omnidirectional antennas [14]. Hence, similar to mobile communication systems, a honeycomb-like structure of communication cells provides the most efficient coverage [15]. Traditionally, square cells are used in WSNs. However, one needs a larger number of square cells to cover an area as compared to hexagonal cells. Assuming that the wireless communication range of the sensors is R, we cover the target field by non-overlapping hexagonal cells with sides $R/2$. If the area of the field is A, then there will be $C = \lceil 8\sqrt{3}A/\left(9R^2\right)\rceil \approx \lceil 1.54A/R^2\rceil$ cells. This choice guarantees that all sensors in a cell are in the communication range of each other. Let n_c be the total number of sensors in each cell. An MKPS is used to establish keys in each cell. We note that since all sensors in a cell are in the communication range of each other, most of the sensors in that cell can establish link keys.

To distribute keys required for the secure communication between adjacent cells, we use a grid-based approach. In this approach, we assign the points on a two-dimensional grid to the cells. In addition, we assign a unique symmetric bivariate polynomial to every cell. To distribute the shares of this polynomial between sensors, we divide the sensors in every cell into equal-size groups. In every cell, the shares of the corresponding polynomial are distributed among the sensors in the groups. Moreover, the sensors of a cell store the shares of the polynomials corresponding to the neighbor cells. As a result, the neighbor cells are able to establish pairwise keys.

The setup algorithm is explained in an algorithmic language in the following.

1) If C is the total number of cells, design a two-dimensional grid with size $m_c \times m_c$, where $m_c := \lceil\sqrt{C}\rceil$. To every cell, assign a unique point (i,j) on the grid where $i,j \in [\,m_c\,]$.
2) Design a pool of m_c^2 symmetric bivariate polynomials $f_{i,j}(x,y) \in \mathbb{F}_{p_c}[x,y]$ with degree t_c in both variables. For all $i,j \in [\,m_c\,]$, assign the polynomial $f_{i,j}(x,y)$ to the cell (i,j).
3) Divide the sensors in the cell (i,j) into G almost-equal-size disjoint groups labeled by $(i,j)_1,\ldots,(i,j)_G$. Let n_g be the maximum number of sensors in every group. For any $g \in [\,G\,]$, store the coefficients of the polynomial $f_{i,j}(g,y)$ in all the sensors in $(i,j)_g$.
4) As shown in Figure 4, assume the six neighbors of the cell (i,j) are (i_ℓ, j_ℓ) for $1 \le \ell \le 6$. Store the coefficients of the six polynomials $f_{i_\ell,j_\ell}(g,y)$ to all the sensors in $(i,j)_g$ for all $g \in [\,G\,]$.

Using this scheme, every sensor stores $7(t_c+1)\log_2 p_c$ bits in addition to $d(t+1)\log_2 p$ bits for the MKPS employed in every cell.

A. Link-Key Establishment

If there is no captured nodes in the network, every two sensors in two adjacent cells are able to establish a direct key using the proposed scheme. Consider two sensors I and I' respectively belonging to two groups $(i,j)_g$ and $(i',j')_{g'}$ in adjacent cells (i,j) and (i',j'). These sensors store the following polynomials in their memories.

$$\begin{aligned} I:&\ \ f_{i,j}(g,y),\ f_{i',j'}(g,y)\\ I':&\ \ f_{i',j'}(g',y),\ f_{i,j}(g',y)\end{aligned}$$

Hence, they are able to calculate the following common keys.

$$k_{I,I',1} = f_{i,j}(g,g') = f_{i,j}(g',g) \tag{10a}$$
$$k_{I,I',2} = f_{i',j'}(g',g) = f_{i',j'}(g,g') \tag{10b}$$

The direct key $k_{I,I'}$ between these two sensors is

$$k_{I,I'} = h(k_{I,I',1}||k_{I,I',2}). \tag{11}$$

V. Evaluation of the LA–MKPS

Since polynomials $f_{i,j}(x,y)$ are bivariate, the recovery of a single polynomial-variable is possible. Therefore, by Lemma 3.1, the scheme is t_c-secure, where t_c is the degree of the polynomials $f_{i,j}(x,y)$ in both variables. Hence, by an analysis similar to the one performed in Section III, we deduce that the probability of the link-key compromise is

$$P_{lkc}^c = \left[1 - \sum_{i=0}^{\min(t_c,G)} \binom{G}{i} p_{cg}^i \left(1-p_{cg}\right)^{G-i}\right]^2, \tag{12}$$

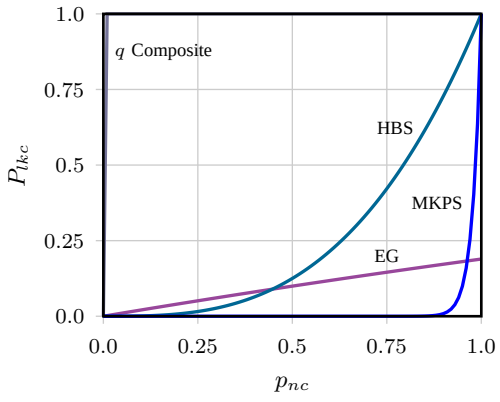

Fig. 3. Probability of the link-key compromise versus the fraction of captured nodes for different schemes. In these curves, $n = 100{,}000$, the node memory is 50, the dimension is $d = 16$, and $q = 3$ in the q-composite scheme.

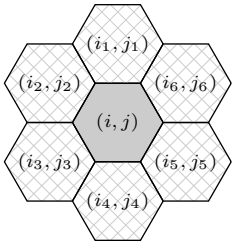

Fig. 4. A hexagonal cell and its six neighbors

where p_{cg} is the probability of compromising a polynomial share. Since every polynomial share is distributed among all the sensors in a group, we have

$$p_{cg} = 1 - (1 - p_{nc})^{n_g}, \tag{13}$$

where p_{nc} is the fraction of captured nodes in the network. If there are n_c sensors in every cell, then $n_g = \lceil n_c/G \rceil$.

In Figure 5, we compare the probability P_{lkc}^c in the location-aware bivariate key pre-distribution (LA–BKPS) of [12] and the proposed LA–MKPS. In these curves, the assumption is that there are $n_c = 100$ sensors in every cell. We note that in the LA–BKPS, every sensor stores five polynomials while in the LA–MKPS, seven polynomials are stored in every sensor. To take into account this difference, we have adjusted the value of t_c in our comparison as $t_c = \lfloor 75/d - 1 \rfloor$, where $d = 5$ in the LA–BKPS and $d = 7$ in the LA–MKPS. As these curves show, the LA–MKPS has a lower probability of the link-key compromise. For example, when 20% of the sensors are captured, about 92% of the link keys in the LA–BKPS are compromised. However, in the LA–MKPS, only 10% of the link-keys are compromised. Another observation is that by increasing the number of sensors n_g in each group, the probability P_{lkc}^c further decreases. This is due to the inverse relationship between n_g and G that affects P_{lkc}^c in (12).

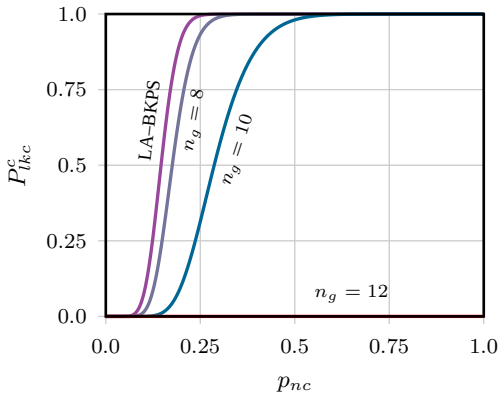

Fig. 5. Probability of the link-key compromise versus the fraction of captured nodes. Parameters are $n_c = 100$, $t_c = 14$ in the LA–BKPS, and $t_c = 9$ in the LA–MKPS. The memory usage of each sensor is 75

VI. Conclusion

In this paper, we proposed a threshold key pre-distribution scheme (KPS) for WSNs. In this scheme, we assign d tuples of nonnegative integers to the sensor nodes as their IDs that are used to distribute shares of multivariate polynomials. After the network deployment, some nodes are able to establish $d-1$ common keys using the shares of polynomials stored in their memories. The secret key between these nodes is a combination of all these $d-1$ keys. Hence, the proposed scheme is, in a sense, a $(d-1)$-composite method. This feature considerably improves the security in the MKPS. Fortunately, this feature is obtained for free with no payoffs such as additional memory. The proposed MKPS has the threshold property, i.e., it remains perfectly secure up to the capture of a certain fraction of sensor nodes.

We also proposed a location-aware version of the MKPS by dividing the terrain into non-overlapping hexagons and uniformly at random distributing nodes. One MKPS layer is used to secure communications inside a cell. For intra-cell communications, a bivariate version of the MKPS is used. This scheme provides perfect connectivity and significantly improves the resiliency of the network against the node capture.

References

[1] T. Arampatzis, J. Lygeros, and S. Manesis, "A survey of applications of wireless sensors and wireless sensor networks," in *Proc. IEEE Int. Symp. Intelligent Control*, vol. 1. Limassol, Cyprus: IEEE, Jun. 2005, pp. 719–724.

[2] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. ACM Conf. Computer Commun. Secur. - CCS'02*. DC: ACM Press, 2002, pp. 41–47.

[3] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. Symp. Security and Privacy*. CA: IEEE Comput. Soc., 2003, pp. 197–213.

[4] R. D. Pietro, L. V. Mancini, and A. Mei, "Random key-assignment for secure wireless sensor networks," in *Proc. ACM Workshop Secur. Ad Hoc Sens. Net. - SASN'03*. NY: ACM Press, 2003, pp. 62–71.

[5] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," in *Proc. ACM Conf. Computer Commun. Secur. - CCS'03*. NY: ACM Press, 2003, pp. 42–51.

[6] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 1, pp. 41–77, 2005.

[7] J. Hwang and Y. Kim, "Revisiting random key pre-distribution schemes for wireless sensor networks," in *Proc. ACM Workshop Secur. Ad Hoc Sens. Net. - SASN'04*. NY: ACM Press, 2004, pp. 43–52.

[8] R. Blom, "Non-public key distribution," in *Proc. Adv. Cryptol. - CRYPTO'82*, D. Chaum, R. L. Rivest, , and A. T. Sherman, Eds. NY: Plenum Publishing, 1982, pp. 231–236.

[9] C. Blundo *et al.*, "Perfectly-secure key distribution for dynamic conferences," in *Proc. Adv. Cryptol. - CRYPTO'92*, ser. Lect. Notes Comput. Sci., E. F. Brickell, Ed., vol. 740. Berlin: Springer-Verlag, 1992, pp. 471–486, carlo Blundo and Alfredo De Santis and Amir Herzberg and Shay Kutten and Ugo Vaccaro and Moti Yung.

[10] H. Pishro-Nik, K. Chan, and F. Fekri, "On connectivity properties of large-scale wireless sensor networks," in *Proc. First Ann. IEEE Commun. Society Conf. on Sensor Commun. and Net.*, Santa Clara, CA, 4-7 October 2004, CD-ROM.

[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 *Proc. IEEE Conf. Comput. Commun. - INFOCOM'04*, vol. 1. NJ: IEEE, 2004, pp. 586–597.

[12] D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," in *Proc. ACM Workshop Secur. Ad Hoc Sens. Net. - SASN'03*. VA: ACM Press, 2003, pp. 72–82.

[13] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware key management scheme for wireless sensor networks," in *Proc. ACM Workshop Secur. Ad Hoc Sens. Net. - SASN'04*. DC: ACM Press, 2004, pp. 29–42.

[14] A. Perrig and J. D. Tygar, *Secure Broadcast Communication in Wired and Wireless Networks*. Boston: Kluwer Academic, 2003.

[15] G. L. Stüber, *Principles of Mobile Communication*, 2nd ed. Boston: Kluwer Academic, 2001.