Journal of Computing and Security

January 2014, Volume 1, Number 1 (pp. 23–38)

http://www.jcomsec.org

An Efficient End to End Key Establishment Protocol for Wireless Sensor Networks

Ali Fanian [a,*]
Mehdi Berenjkoub [a]
[a] *Department of Electrical and Computer Engineering, Isfahan University of Technology (IUT), Isfahan, Iran.*

ARTICLE INFO.

Article history:
Received: 05 November 2012
Revised: 06 April 2013
Accepted: 28 June 2013
Published Online: 20 December 2013

Keywords:
Wireless Sensor Networks, Key Management, Network Security, Random Key Pre-distribution, Symmetric Polynomial, Deployment Knowledge, Combinatorial Design.

ABSTRACT

Sensor networks are eligible candidates for military and scientific applications such as border security and environmental monitoring. They are usually deployed in unattended or hostile environments; therefore, security is a major concern with these networks. A fundamental requirement is the capability to establish pairwise keys between sensors. Many key establishment protocols have been proposed to address the security issues in wireless sensor networks. However, most of these protocols have security and/or performance restriction. In this article, we propose a new key establishment protocol based on symmetric polynomials and random key pre-distribution. In our protocol, contrary to others, we use several symmetric polynomials to generate polynomial shares for a group of sensors, and the distribution of polynomial shares to each sensor is done using a combinatorial design. Since a limited number of shares are generated from a symmetric polynomial, the polynomial degree is very low. As a result, the common key between sensors can be generated without imposing significant overhead to them. Further, in the proposed protocol, key establishment between near sensors is provided via symmetric polynomials, while key establishment between far sensors is accomplished via random key pre-distribution. Using these two techniques simultaneously allows end to end key establishment between every pair of sensors with reasonable overhead.

1 Introduction

Due to the technological achievements in recent decades, wireless networks are now widely deployed and highly developed. According to the application, there are many types of wireless networks, including sensor networks. Sensor networks are usually constructed from a large number of limited resource sensors, which are used to collect information from the surrounding environment [1–3]. These networks are suitable for many purposes such as border security, military target tracking, and scientific research in dangerous environments [4–6]. Sensors commonly communicate with each other via a wireless channel. Since the sensors may be used in hostile environments, especially in military usage, security is a very important issue with these networks. Therefore, security services such as encryption and authentication should be used. Since authentication and encryption use a common key between nodes, key management is a

* Corresponding author.
Email addresses: a.fanian@cc.iut.ac.ir (A. Fanian), brnjkb@cc.iut.ac.ir (M. Berenjkoub)
ISSN: 2322-4460

critical aspect of network security. In the literature, key management protocols are based on either symmetric or asymmetric encryption techniques. Due to resource limitations in the sensors, protocols based on asymmetric cryptography are not suitable [6, 7]. To date, many different methods have been proposed for key management in sensor networks [8–16]. Some of these distribute sensors in the network uniformly [8–11] so sensors can reside at any point in the network with equal probability. Key establishment with these methods presents both security and performance challenges. Others have tried to improve performance using node deployment knowledge. This knowledge is used in the pre-deployment stage to arrange the sensors into groups [12–16]. Some of the proposed protocols are based on random key pre-distribution. In these schemes, one key may be used for several pairs of sensors, so perfect security cannot be achieved. With perfect security, the security between two uncompromised sensors must be guaranteed despite other sensors being compromised. Some protocols are based on threshold cryptography. While perfect security can be achieved with this technique, the memory usage and processing overhead are very high. Further, in most of the proposed protocols, end to end key establishment between every pair of sensors creates significant communication overhead. In this article, an efficient end to end key establishment protocol (ETE) is proposed which uses both symmetric polynomial and random key pre-distribution methods. This improves the security, and also the scalability so that the sensor overhead due to the protocol is not excessive. A preliminary version of this article was presented in [17], however in this article the proposed key establishment for near and far sensors are considered simultaneously. Also the technical correctness of the proposed protocol is precisely evaluated.

The features of the proposed protocol are as follows:

- The polynomial shares are distributed in groups based on combinatorial design theory.
- Shared keys between sensors within each group are directly established based on their polynomial share derived from a symmetric polynomial.
- Keys between two sensors which are not in the same group but are near to each other are indirectly generated by intermediate nodes called agents. Keys are sent from the source sensor to the destination sensor via a secure path generated by agent sensors.
- Key establishment between far sensors is provided by random key pre-distribution.
- End to end key establishment between every pair of sensors in the network can be achieved without significant overhead.
- The proposed protocol can provide connectivity and resistance against key exposure for a large network with relatively low resource consumption.
- Perfect security can be supported in the proposed protocol.
- The proposed protocol is more efficient in terms of processing overhead than other schemes.

The rest of the article is organized as follows. Section 2 reviews some required primitives including related work. Details of our key establishment protocol are given in Section 3. Performance evaluation and security analysis of the proposed protocol are presented in Section 4. Finally, some conclusions are given in Section 5.

2 Preliminaries

In this section, we present authentication using symmetric polynomials and review related results in the area.

2.1 Authentication Using Symmetric Polynomials

A symmetric polynomial [11], [18, 19] is a t-degree $(K+1)$-variate polynomial defined as follows

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

All coefficients of the polynomial are chosen from a finite field F_q, where q is a prime integer. The polynomial f is a symmetric polynomial in the following sense [11]

$$f(x_1, x_2, \ldots, x_{k+1}) = f(x_{\partial(1)}, x_{\partial(2)}, \ldots, x_{\partial(k+1)}) \quad (2)$$

This means that for any permutation, ∂, we obtain the same polynomial. Every node using the protocol based on a symmetric polynomial takes identities $(I_1, I_2, , I_K)$ from the key management center, and stores these in memory. The key management center must also compute a polynomial share for each node using its identities and the symmetric polynomial. The coefficients $b_{i_{K+1}}$ stored in node memory as the share of the polynomial coefficients are obtained using the following expression

$$f(_u(x) = f(I_1, I_2, \ldots, I_k, x) = \sum_{i_{K+1}=1}^{t} b_{i_{K+1}} x^{i_{K+1}} \quad (3)$$

Every pair of nodes with only one mismatch in their identities can establish a common key. Suppose the

identities of nodes u and v have one mismatch in their identities given by $(c_1, c_2, \ldots, c_{i-1}, u_i, c_{i+1}, \ldots, c_K)$ and $(c_1, c_2, \ldots, c_{i-1}, v_i, c_{i+1}, , c_K)$, respectively. In order to compute a common key, node u takes v_i as the input and computes $f_u(v_i)$, and node v takes u_i as the input and computes $f_v(u_i)$. Due to polynomial symmetry, both nodes compute the same key. In [11], it was shown that in order to maintain perfect security in the sensor network, the polynomial degree must satisfy the presented inequalities in 4 and 5.

$$0 \leq N_i - 1 \leq t \tag{4}$$

$$N_i \times \sqrt[K+1]{\frac{K(K+1)!}{2}} \leq t \tag{5}$$

where N_i is the number of nodes in group i and $i = 1, 2, \ldots, K$

2.2 Combinatorial Design Theory

Combinatorial design theory is the study of arranging elements of a finite set into patterns such as subsets or arrays according to specified rules. For instance, one such design is a Balanced Incomplete Block Design ($BIBD$). A $BIBD$ is an arrangement of a finite set S of v distinct object into a collection B of b blocks such that every block has k distinct objects. Each object occurs in r different blocks, and each pair of S certainly occurs in λ blocks. The design can be expressed as (v, b, r, k, λ) where $\lambda(v-1) = r(k-1)$ and $bk = vr$[20, 21]. A $BIBD$ is called a symmetric $BIBD$ when $b = v$ and $r = k$. In this case, each block has $k = r$ objects and each object is in $r = k$ blocks. For example consider a $(v, b, r, k, \lambda) = (7, 7, 3, 3, 1)$ symmetric design. Let $S = \{1, 2, 3, \ldots, 7\}$, so the set S has 7 objects, and there are 7 blocks. Each block has 3 objects, and each object occurs in 3 blocks. Every pair of distinct objects occurs in only one block. Consequently, every pair of blocks intersects in only one object. Using a construction algorithm, the blocks are: {1,2,3}, {1,4,5}, {1,6,7}, {2,4,6}, {2,5,7}, {3,4,7}, {3,5,6}. A subset of symmetric $BIBDs$ is called Finite Projective Planes. These planes consist of a finite set P points and a set of subsets of P, called lines. For an integer q, $(q2)$, the finite projective plane has four properties: 1) every line contains q+1 points; 2) every point occurs in $q+1$ lines; 3) there are q^2+q+1 points as objects; 4) there are q^2+q+1 lines as blocks. Therefore, a finite projective plane of order q is a symmetric design with parameters $(q^2+q+1, q^2+q+1, q+1, q+1, 1)$ [20, 21].

2.3 Related Work

Due to resource constraints in the sensors, key management is not a trivial task in WSNs. Thus, many key establishment protocols have been proposed to address the security issue in WSNs. Eschenauer et al. [8] proposed a random key pre-distribution scheme. Before sensor deployment, keys from a large key-pool are selected randomly and stored in the sensors. After deployment, with some probability they can establish a key between each other. If there is no common key between two sensors, a common key can be generated through an intermediate node which has common keys with both. Zhou et al. [11] proposed a key management protocol based on symmetric polynomials. In this scheme, a t-degree $(K+1)$-variate symmetric polynomial is selected and each sensor has a K-tuple of identities. Then the polynomial share for each sensor is computed using these identities and the symmetric polynomial. The polynomial share and sensor identities are stored in the sensor. If the identities of two sensors differ in only one dimension, they can produce a common key. Liu et al. proposed another key management protocol based on symmetric polynomials [10]. In their approach, a grid based model is used to calculate and distribute the sensor polynomial shares. The N sensors in the network form an $m \times m$ matrix ($N = m^2$). For each row and column of this matrix a t-degree bi-variate symmetric polynomial is selected. A sensor in position (i, j) of the matrix computes polynomial shares $f_j^c(x, j)$ and $f_i^r(i, y)$ from symmetric polynomials $f_j^c(x, j)$ and $f_i^r(i, y)$, respectively. Therefore, if two sensors are in the same row or column, they can compute a common key directly. Otherwise, they have to employ another sensor to compute this key. Note that the sensor deployment model in this method does not follow any distribution, so the sensors are distributed randomly in the network.

Some WSN key management schemes use deployment knowledge to improve the protocol performance. In [13], [15] it was shown that if deployment knowledge is used before the sensors are deployed, the efficiency of the protocol can be increased. Du et al. [13] proposed a key management protocol based on the approach in [8] which uses deployment knowledge during key distribution. This knowledge is modeled using a Gaussian probability distribution function (pdf). Conversely, methods which do not use deployment knowledge follow a uniform model, so the sensors can reside at any point in the network with equal probability. In [13] the network area is divided into square cells and each cell is considered the location for one group of sensors. Then the key-pool is divided into sub key-pools (equal to the number of cells) such that each sub key-pool has some key correlation with its neighboring sub key-pools. Each sensor in a cell has m random keys selected from its corresponding sub key-pool. Therefore, using deployment knowledge in the pre-distribution of keys allows for keys to be se-

lected from a smaller key-pool. This can improve the protocol performance, especially in large networks.

Lin et al. [12] proposed a key management protocol called LPBK in which the network area is divided into square cells. Each cell has a specific symmetric polynomial which is used to compute a polynomial share for the sensors in the corresponding cell and the four adjacent vertical and horizontal cells. In this scheme, each sensor stores five polynomial shares in its memory. Zhou et al. [14] proposed a key management protocol called LAKE which is based on symmetric polynomials and deployment knowledge. Similar to the previous scheme, the network is divided into square cells and each sensor is allocated to a cell. The polynomial in this method is a t-degree tri-variate symmetric polynomial. Each sensor in this protocol has a pair of identities (n_1, n_2) where n_1 represents the cell identity and n_2 represents the sensor identity within the cell. Each sensor receives a polynomial share based on its identities. After network deployment, sensors with just one mismatch in their identities can directly compute a common key.

Yu et al. [16] proposed another key management protocol based on an approach by Blom [22]. Blom developed a protocol that allows each pair of nodes to establish a common key. With this scheme, if no more than t nodes are compromised, all common keys of the uncompromised nodes remain secure. A $(t+1) \times N$ matrix G is defined as public information where N is the size of the network. During the key generation phase for the sensor nodes, the key management server creates a secure random $(t+1) \times (t+1)$ symmetric matrix, D. Then the server calculates an $N \times (t+1)$ matrix $A = (D \times G)^T$, where T denotes transpose. Since D is symmetric, $K = A \times G$ is also symmetric, so that $K_{ij} = K_{ji}$. In Bloms scheme, K_{ij} is used as a common key between nodes i and j. Hence, nodes i and j should be able to compute K_{ij} and K_{ji}, respectively. Node i stores the ith row of A and the ith column of G. Hence, if nodes i and j want to establish a common key, they should exchange their columns of G and then compute $K_{ij} = K_{ji}$ using their private row of A. In the Yu et al. scheme, the network area is divided into hexagonal cells and the information from the G and D matrices is stored in the sensors based on deployment knowledge. A secret matrix A_i (equivalent to matrix D in the Blom method), is allocated to cell i. The sensors in the cell are each assigned a row from A_i. Hence, the sensors belonging to a cell can produce a common key directly. To generate a common key between sensors belonging to different cells, another secret matrix, B, is used that is common to sensors in neighboring cells. These matrices are allocated to the cells using two parameters b and w. Parameter b indicates the number of B matrices assigned to a group, and w is the number of rows chosen by a sensor. The analysis in [16] shows that the best results are obtained with w=2 and b=2. In this method, the cells are divided into two categories, base cells and normal cells. Base cells are not neighbors to each other, but every normal cell is a neighbor of two base cells and four normal cells. To produce a common key between sensors in neighboring cells, a confidential matrix B is allocated to each base cell together with its six neighbors. Bloms scheme is used with this matrix to provide information to the sensors. Therefore, the sensors in the seven neighboring cells can produce a common key directly. Since every normal cell is neighbor with two base cells, their sensors receive information from two B matrices. Although this scheme has connectivity near to one, the memory consumption is extremely high. Du et al. [15] proposed another key management protocol using Blom scheme. In this scheme, many pairs of matrices G and D, called the key space, are produced, and some key spaces are assigned to each cell. Adjacent cells have some common key space as in [13], where adjacent cells have correlation between their sub key pools. Each sensor selects key spaces randomly, and according to Blom scheme, the required information is stored in the sensors. As a result, sensors with the same key space can produce a common key. As in [13], two vertical or horizontal neighboring cells have common key spaces, and two diagonal neighboring cells have common key spaces, where is the number of key spaces assigned to a cell.

In [23], a hybrid key establishment protocol for low resource wireless sensor networks based on Blom scheme and random key pre-distribution was proposed. In this protocol, the network area is divided into non-overlapping hexagonal cells, and key establishment between sensors in a cell is provided by Blom scheme. Key establishment between sensors residing in neighboring cells is provided via pre-distributed random keys.

Kavitha et al. [24] proposed a hybrid key distribution scheme that combines both probabilistic and deterministic approaches in order to utilize the advantages of both methods. In this scheme, some keys are assigned to some sensors based on a deterministic approach, and some keys are distributed to sensors randomly. Mutual Octagonal Latin Squares (MOLS) are used for deterministic assignment. In [25], a key management protocol called SKEP based on symmetric polynomials was proposed. In this protocol, a symmetric polynomial was used to generate some shares between sensors belonging to a group. Although, in [25] sensors can simply determine whether or not they can generate a common key with other sensors, but the local connectivity of SKEP is about 0.5. More-

over, if the number of sensors belonging to a group increases, in perfect security conditions the computation overhead become considerable.

3 The End to End Key Establishment Protocol (ETE)

We propose a new key management protocol called ETE. Both symmetric polynomials and random key pre-distribution are used to improve efficiency. Similar to other schemes [12–15], key information is distributed to the sensors during the pre-deployment stage. Once the sensors have been deployed, they can produce a common key either directly or indirectly. Due to the use of two methods in ETE, two types of information must be stored in the sensors. One is the sensor polynomial shares generated using the symmetric polynomials and finite projective planes, while the other is a set of random keys.

There are two types of key generation in ETE. In the first type, a common key between near sensors is generated via their polynomial shares. In the second type, a common key between far sensors is generated using the pre-distributed random keys. Since in this case a key may be selected by several sensors, the common key between two far sensors may also be used by other pairs of sensors. Conversely, the common key between near sensors is unique. As we will show, end to end key establishment between every pair of sensors can be supported without significant overhead.

A common key between far sensors may be required in situations where environmental information is transmitted to servers far from the network area. Hence, the information must be relayed between sensors to reach the servers. Thus an authenticated route must be established between sensors or between sensors and servers by using a secure routing protocol such as Karlof [26], Ariaden [27] and endairA [28]. In a secure routing protocol, the intermediate sensors which establish the route must all be authenticated. This is necessary to prevent an adversary from deceiving the server with false information. In other words, if the intermediate sensors rely only on authenticated information from the other sensors, adversaries or malicious sensors cannot send false information through the network. To achieve this security, each sensor must be able to authenticate the other sensors. Therefore, the sensors should be able to establish a common key with every other sensor. Thus in the proposed protocol, as in [13, 14], a goal is the capability of generating a key between every pair of sensors. With other schemes, intermediate nodes must be used to establish a common key between far sensors [12], [16]. Because the number of intermediate nodes may be large, the possibility of common key exposure is high, and the use of these nodes increases the communication and computational overhead. Key generation between near sensors will, however, occur more frequently than between far sensors, so we use different key generation processes for near and far sensors to achieve high performance. Moreover, in ETE, we use deployment knowledge to distribute key information among the sensors. ETE consists of three phases: key and polynomial share pre-distribution, direct key establishment and indirect key negotiation. We next introduce the models used in ETE, and then discuss the key management phases in detail.

3.1 ETE Models

In this section we introduce both the security and network models used in ETE.

3.1.1 Deployment Model

As shown in [13], [15] deployment knowledge can be used to improve network connectivity and increase key management performance. As mentioned before, the key management schemes based on deployment knowledge such as [13–16] use a Gaussian distribution for the distribution of sensors in the network. In ETE, the network area is divided into non-overlapping hexagonal cells, and the sensors are allocated in groups to these cells. The center of a cell is defined as the deployment point of the sensors allocated to that cell. Figure 1 shows the division of the network into hexagonal cells. Each cell in ETE has a pair of identities (i, j) which is the cell position. Using two-dimensional Cartesian coordinates and assuming that the deployment point of cell $C_{i,j}$ is (x_i, y_i), the pdf of the sensor resident points can be formulated as

$$f_k^{ij}(x, y|k\epsilon C_{i,j}) = f(x - x_i, y - y_j) = \frac{1}{2\pi\sigma^2} e^{-[(x-x_i)^2+(y-y_i)^2]/2\sigma^2}$$

where

$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{-[x^2+y^2]/2\sigma^2} \quad (6)$$

Assuming the same pdf for all groups, we use $f_k(x, y|k\epsilon C_{i,j})$ instead of $f_k^{ij}(x, y|k\epsilon C_{i,j})$. As shown in [15], the distance between the deployment point and resident point of a sensor is less than 3σ with probability 0.9987, which σ is the standard deviation.

3.1.2 Security Model

The security model used in ETE is based on two mechanisms: symmetric polynomials and random key distribution. We introduce these mechanisms below.

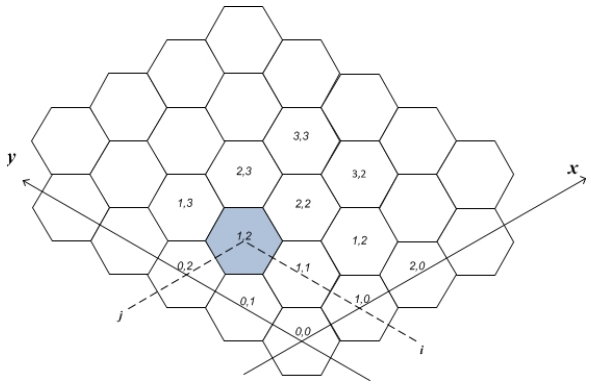

Figure 1. Deployment points for a two-dimensional sensor network

1- Generating the Polynomial Share based on Symmetric BIBD

In the key establishment protocols based on threshold cryptography such as Blom scheme or symmetric polynomials, some shares are generated from a secret for many sensors in a group. Hence, to achieve perfect security the threshold must be set very high. As we show in the processing analysis section, increasing the threshold causes a significant increase in processing overhead. Thus in the proposed protocol, contrary to other schemes, we use several symmetric polynomials instead of one, but each symmetric polynomial generates a limited number of polynomial shares for sensors. In other words, each sensor in a group has several polynomial shares, but the polynomial degrees are very low. Increasing the number of symmetric polynomials for a group may create some difficulties in the key establishment protocol. For example, some sensors may have more than one polynomial share from the same symmetric polynomial, or some sensors may have no shares from the same polynomial. Therefore in ETE, the distribution of polynomial shares is very critical to the protocol performance, so we use combinatorial design theory to distribute polynomial shares in a group.

As mentioned previously, a symmetric BIBD can be represented using the parameters (v, b, r, k, λ). Since in a symmetric BIBD $v = b$ and $r = k$, we can represent it by (v, k, λ), where v is the number of distinct objects, k is the number of distinct object in a block and λ is the number of times a pair of objects occurs in the blocks. We use a symmetric BIBD for distributing symmetric polynomial shares to the sensors.

In the proposed protocol, a symmetric polynomial is as an object in the symmetric BIBD and each sensor is a block. A symmetric polynomial is used to generate a common key between two sensors, so the intersection of the assigned symmetric polynomials for two sensors should be one. As a result, we assume λ=1. Thus, we consider a finite projective plane of order q with parameters $(q^2 + q + 1, q + 1, 1)$, and q should be chosen such that $q^2 + q + 1$ is greater than the number of sensors in a group, Ng. As mentioned in Section 3.1.1, the network area is partitioned into hexagonal cells, and the cells are divided into base cells and normal cells. Base cells are not neighbors, but normal cells are neighbors with two base cells and four other normal cells. Each normal cell is divided into two virtual regions. As shown in Figure 2, a group consists of a base cell and virtual regions in the six neighboring cells. In this figure, cells (i, j), $(i, j + 2)$ and $(i + 2, j + 2)$ are base cells and G1, G2 and G3 are groups and other cells are normal cell. The half of each normal cell is virtual region. Each virtual region belongs to one group, so q should be chosen according to this group size. To construct a symmetric design, we use $(q - 1)$ Mutually Orthogonal Latin Squares (MOLSs). A MOLS is a $q \times q$ array such that each of the q symbols occurs exactly once in each row and column [20, 21]. For example L_1 and L_2 below are two MOLS of order 3.

$$\mathbf{L1} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

$$\mathbf{L2} = \begin{pmatrix} 1 & 3 & 2 \\ 3 & 2 & 1 \\ 2 & 1 & 3 \end{pmatrix}$$

According to [20, 21], a symmetric BIBD with $(q^2, q, 1)$ is called affine plan. Therefore q^2 blocks of an affine plane can be constructed using the generated MOLS and then the blocks of the projective plane can be constructed from this affine plane. The blocks of the finite projective plane from L_1 and L_2 are the following.

$$\begin{aligned}
A_1 &= (1,1),(2,2),(3,3),(4,1)\\
A_2 &= (1,3),(2,1),(3,2),(4,1)\\
A_3 &= (1,2),(2,3),(3,1),(4,1)\\
A_4 &= (1,1),(2,3),(3,2),(4,2)\\
A_5 &= (1,3),(2,2),(3,1),(4,2)\\
A_6 &= (1,2),(2,1),(3,3),(4,2)\\
A_7 &= (1,1),(1,2),(1,3),(4,3)\\
A_8 &= (2,1),(2,2),(2,3),(4,3)\\
A_9 &= (3,1),(3,2),(3,3),(4,3)\\
A_{10} &= (1,1),(2,1),(3,1),(4,4)\\
A_{11} &= (1,2),(2,2),(3,2),(4,4)\\
A_{12} &= (1,3),(2,3),(3,3),(4,4)\\
A_{13} &= (4,1),(4,2),(4,3),(4,4)
\end{aligned}$$

Now we have 13 blocks that represent the symmetric block design. Note that the symmetric design conditions are satisfied by these blocks, and it has parameters $(q^2+q+1, q+1, 1)$. In our protocol, we use the finite projective plane to distribute polynomial shares to sensors in a group. As mentioned previously, the groups of sensor are constructed and a finite projective plane is used for a group. Since objects are symmetric polynomials and blocks are sensor nodes, we must generate $q^2+q+1 \geq 4N_c$ symmetric polynomials for each group, where N_c is the number of sensors in a cell. Each symmetric polynomial is a t-degree bi-variate symmetric polynomial, given by $f(x_1, x_2) = \sum_{i_1=1}^{t} \sum_{i_2=1}^{t} a_{i_1,i_2} x_1^{i_1} x_2^{i_2}$. Each sensor in ETE has three identities (i, j, k). The first two identities (i, j) are the deployment point of the sensor and the last identitie (k) is the unique sensor identity in the cell. Based on the finite projective plane, $q+1$ symmetric polynomials are assigned to a sensor. For each symmetric polynomial, the sensor polynomial share $b_{i_2}^v$ is generated as follows

$$f_k^v(x) = f_{i,j}^v = \sum_{i_1=1}^{t} \sum_{i_2=1}^{t} a_{i_1,i_2} x_1^{i_1} x_2^{i_2} \quad b_{i_2}^v \sum_{i_1=1}^{t} a_{i_1,i_2} k^{i_1}$$

where $v \epsilon \{1, 2, \ldots, q^2+q+1\}$.

Since every pair of sensors in a group have a polynomial share from the same symmetric polynomial, they can directly establish a common key. However, sensors belonging to different groups cannot establish a common key directly or even indirectly. This creates isolated components within the network. Thus in the proposed protocol, we use another finite projective plane for each normal cell. As shown in Figure 2, half of the sensors belonging to a normal cell are in one group, and the others belong to a second group. Hence, two sensors in a normal cell that belong to different groups cannot establish common key. Using the new finite projective plane, we want to generate polynomial shares for sensors that belong to a normal cell. This requires the generation of $q^2+q+1 \geq N_c$ symmetric polynomials similar to those for the first finite projective plane. Then $q+1$ polynomial shares are computed and assigned to each sensor.

2- Random Key Selection

In methods based on random key pre-distribution, some keys are selected randomly from a key-pool with size S and assigned to the sensors. As mentioned in Section 2.2, in [13] the global key-pool is divided into sub key-pools and each sub key-pool is assigned to a cell so that neighboring cells have some correlation in their sub key-pools. By correlation we mean that the intersection of the two sets is nonzero. Since random keys are only used for far sensors in ETE, contrary to [13], neighboring cells should not have any correlation

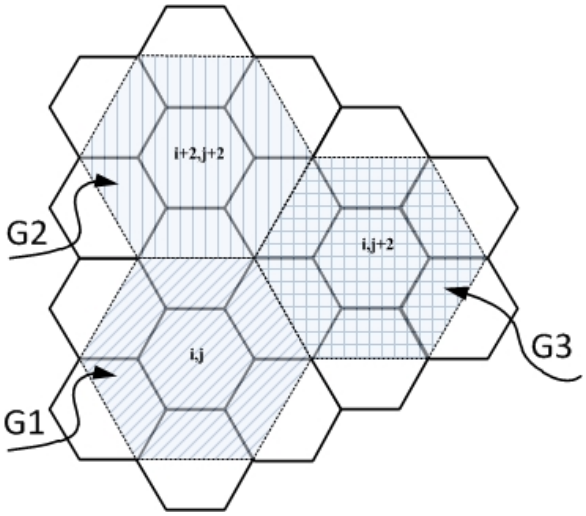

Figure 2. The group model in the proposed protocol

in their sub key-pools. Thus, sensors in base cells should select their random keys from separate sub key pools. In other words, in the proposed protocol, sensors in a normal cell receive two sets of polynomial shares from two finite projective planes, but the sensors in base cells receive only one set of polynomial shares from one projective plane. Hence, the memory used in base cell sensors is less than that used in normal cell sensors, so we only assign random keys to sensors in base cells.

According to the proposed model, the global key pool is divided into N_c sub key-pools and each sub key-pool has S/N_c keys, and each sensor in a base cell selects keys from a specific sub key pool based on its identity, so all sensors in base cells with the same identity select random keys from the same sub key pool. This ensures that two sensors in the same cell do not select keys from the same sub key pool. In addition, the sub key-pool assigned to a given sensor can easily be determined. This greatly simplifies key discovery, as will be described later.

3.2 Key Establishment by Near Nodes

Two nodes that want to produce a common key can estimate their distance by knowing each others identities. This distance can be used to decide whether nodes are near or far away. The real distance between sensors is difficult or impossible to compute, so we use logical distance instead of real distance. The logical distance between two sensors is the distance between their deployment points. The sensor deployment point,(i, j), is specified by the sensor identity (i, j, k). If this logical distance is less than a threshold dTl, we assume the sensor nodes are near to each other and a common key can be obtained using polynomial shares. Otherwise, we assume the nodes are far from each other

and a common key will be established using the pre-distributed random keys. Due to the distribution of random keys in ETE, the logical distance between two cells using the same sub key-pool is three. Therefore, dTl should be less than this distance, so two sensors are said to be near if their logical distance is less than three. Key establishment between sensors falls into three categories as described below.

3.2.1 Direct Key Generation

Sensors belonging to the same cell can produce a common key directly. Since the real distance between them may be greater than the wireless transmission range, intermediate sensors may be required to relay the information. Some sensors located in neighboring cells may belong to the same group. They can also produce a common key directly. In fact, some sensors in all six normal cells adjacent to a base cell will have suitable polynomial shares. Note that the number of sensors belonging to a group is $4 \times N_c$. According to the model in Section 3.1.2, two sensors can easily determine if they are in the same group. If they belong to the same group, they can produce a common key by exchanging their unique identities as shown below

$$f_1(k_2) = \sum_{j=0}^{t} b_{j1} k_2^j = K_{1-2}$$
$$, f_2(k_1) = \sum_{j=0}^{t} b_{j2} k_1^j = K_{2-1} \quad (7)$$

3.2.2 Indirect Key Generation between Two Sensors

Sensors belonging to two distinct cells may reside adjacent to each other. If they cannot establish a common key directly, a proper agent node must be used. A proper agent is a node that has common keys with both of them. For example, in Figure 2, suppose that node u from cell C(i,j) belonging to group G1 wants to establish a common key with node v from cell $(i{+}1, j{+}2)$ belonging to G2. First, node u must find two agent sensors in cell $(i{+}1, j{+}1)$, one belonging to G1 and another belonging to G2. Then node u generates K_{uv} as a common key and sends it to the agent nodes via a secure channel. The agent nodes receive this key and send it to v via a secure channel. Note that there are $N_c/2$ candidates for each agent node, where N_c is the number of sensors in a cell. If the deployment distance of two sensors belonging to two cells is more than 3 cells, random key pre-distribution is used for indirect key generation. Otherwise, since these sensors have not any polynomial share from similar symmetric polynomial, they have to use some agent nodes to establish common key. Therefore, in this situation, the key establishment imposes significant overhead to sensor nodes. Note that if the distance between the resident points of the agents is less than the wireless transmission range, they will communicate with each other directly. Otherwise, a routing protocol can be used between them as in [8–15].

3.3 Key Generation between Far Sensors

If two sensors are far from each other, generating a common key using agents and their polynomial shares can be time consuming, and creates a security risk if nodes have been compromised. In this case, instead of establishing a long path of agents, ETE uses the pre-distributed random keys stored in base cell sensors. As mentioned previously, all sensors in a base cell have some random keys selected from separate sub key-pools. If two sensors select their random keys from the same sub key-pool, they can establish a common key. Otherwise, another sensor must be used. Due to the proposed random key pre-distribution, sensors with the same identity belonging to different base cells have selected keys from the same sub key-pool. Hence, there are N_c sensors in each cell which can be used as an agent sensor to establish a common key between two sensors that reside in far cells. If two sensors belonging to normal cells want to establish a common key, they must select two sensors from base cells with the same identity as agent nodes. Therefore, the number of agents required is two. If there is no common key between these sensors, another pair of sensors in the base cells must be selected to reinitiate the process of key establishment. This is repeated until a common key is established between agents in the two cells.

4 Security Analysis and Performance Evaluation

In this section the security analysis and performance evaluation of the proposed protocol are presented and compared with similar protocols. Based on the discussion in Section 2.2, the best candidates for comparison with ETE are LPBK [12], LAKE [14], and the protocols by Yu et al. [16] and Du et al. [15].

4.1 Network Configuration

The network parameters are as follows:

- The number of sensors in the network is N = 10000
- The network area is $1000 \times 1000 m^2$
- The sensor locations have a two dimensional Gaussian distribution with standard deviation
- The wireless transmission range is 40 m

- The number of sensors deployed in each cell is 100

4.2 Determining the Polynomial Degree

To satisfy the perfect security condition, the security between two uncompromised sensors must be guaranteed despite the existence of compromised sensors. As mentioned in Section 3.2, in ETE direct key generation via a symmetric polynomial is limited to $q+1$ sensors. Thus in this case, if an adversary wants to obtain the common key between two sensors, it must compromise a number of sensors who have polynomial shares generated from the same polynomial. This requires that an adversary produce the polynomial share of either sensor. To do this, an adversary must solve an equation with $(t+1)$ unknowns, i.e., the sensor polynomial share coefficients b_{i_2}. By compromising X_i sensors in the same group as sensor i, an adversary can produce the following set of equations

$$\begin{aligned} f_0(k_i) &= \textstyle\sum_{j=0}^{t} b_{0j}k_i^j = K_{0-2} \\ f_1(k_i) &= \textstyle\sum_{j=0}^{t} b_{1j}k_i^j = K_{1-2} \\ &\vdots \\ f_{X_i}(k_i) &= \sum_{j=0}^{t} b_{X_i j}k_i^j = K_{X_i-2} \end{aligned} \tag{8}$$

Since the polynomial share of the ith sensor has $t+1$ coefficients, if X_i is less than $t+1$, an adversary cannot obtain this share. Therefore, if the number of sensors who have polynomial shares from a symmetric polynomial, N_i, is less than $t+1$, an adversary cannot compromise the common key between two uncompromised sensors, even if all other sensors are compromised.

For key management protocols based on symmetric polynomials and deployment knowledge, we must compute N_i. In the proposed protocol, as shown in Figure 2, each group includes all sensors belonging to a cell and some sensors in the six neighboring cells. However, in ETE N_i is equal to $q+1$ where q satisfies the condition $q^2+q+1 \geq 4N_c$. With LPBK [12], the sensors in a cell along with its four horizontal and vertical neighbor cells have a polynomial share from the same symmetric polynomial. Thus N_i is equal to $5N_c$. In [14] each sensor has a polynomial share which can be used to directly produce a common key with all other sensors in its own cell and N_c sensors in other cells. Therefore N_i is equal to $2N_c$. To achieve full security with these methods, the polynomial degree must satisfy

$$N_i - 2 \leq t \tag{9}$$

In Table 2, the required polynomial degree t is given for different protocols based on symmetric polynomials.

Table 1. The Minimum Polynomial Degree Required to Achieve Perfect Security

Protocol	Polynomial Degree
LPBK [12]	498
LAKE [14]	198
ETE	19

4.3 Network Connectivity

Network connectivity is the ability to generate a common key between sensors in the network. In [13], and [14] network connectivity is classified into two types: local connectivity and global connectivity. Local connectivity considers the probability of directly producing a common key between adjacent sensors. Global connectivity is defined as the ratio of the number of nodes in the largest isolated component to the size of the network. For example, if the global connectivity is 0.98, only two percent of the sensors cannot be accessed by the largest component. A network may have high local connectivity but there are some isolated components in the network. We define another type of connectivity called remote connectivity. Remote connectivity is the probability of producing a common key between far sensors. Note that if there is a common key between far sensors, intermediate sensors can act as routers so there is no need to establish a long key path.

4.3.1 Local Connectivity

Local connectivity is the probability of producing a common key between two adjacent sensors. Due to the polynomial share key generation mechanism in ETE, the probability of key generation between two adjacent sensors belonging to same group is one. However, some adjacent sensors do not belong to the same group, so they cannot establish a common key directly. To determine the local connectivity, we must determine the probability of producing common key for adjacent sensors. In [13, 15] the probability of local connectivity is defined as

$$P_{L-C} = Pr(B(n_i, n_j)|A(n_i, n_j)) \tag{10}$$

where $B(n_i, n_j)$ is the event of directly producing a common key between sensors n_i and n_j, and $A(n_i, n_j)$ is the event that two nodes are adjacent. As shown in [13, 15], P_{L-C} is given by

$$
\begin{aligned}
P_{L-C} &= p_1/p_2 \\
p_1 &= p(B(n_i,n_j)\ and\ A(n_i,n_j)) \\
&= \sum_{j\epsilon\psi}\sum_{i\epsilon\psi} Pr(B(u,v)\ and\ A(u,v)|u\epsilon C_i\ and \\
&\qquad v\epsilon C_j)\times Pr(u\epsilon C_i\ and\ v\epsilon C_j), \\
p_2 &= \sum_{j\epsilon\psi}\sum_{i\epsilon\psi} Pr(A(u,v)|u\epsilon C_i\ and\ v\epsilon C_j)\times \\
&\qquad Pr(u\epsilon C_i\ and v\epsilon C_j) \qquad (11)
\end{aligned}
$$

The interested reader is referred to [13, 15] for details. In this article, we use Equation 11 to compare the local connectivity with all methods. The value of $A(u,v)$ depends on the cell size, the number of sensors in each cell, the wireless transmission range, and the standard deviation of the Gaussian distribution for the sensors. These parameters are related to the network configuration introduced in Section 4.1. With ETE, the probability of common key generation between two sensors in cells i and j, is given by

$$
P(B(u,v)|u\epsilon C_i\ \ and\ \ v\epsilon C_j) =
\begin{cases}
1 & \text{if } i=j, \\
0.5 & \text{where } i,j \text{ denotes a normal cell and its adjacent base cell,} \\
0.25 & \text{where } i,j \text{ denotes two adjacent normal cells,} \\
1-\binom{Sc}{Sc-r}/\binom{Sc}{r} & \text{where } i,j \text{ denotes two base cells,} \\
0 & \text{others}
\end{cases}
\qquad (12)
$$

where Sc is the sub key pool size and r is the number of random keys selected by each sensor in a base cell. Local connectivity with the other methods was computed by considering the probability of generating a common key between sensors and Equation 11. The results are shown in Table 3. With the scheme in [13], each cell has a sub key-pool with Sc keys. Some keys in a sub key-pool also exist in the sub key-pools of neighboring cells. In [13, 15], it is assumed that each cell shares 15% of its keys with each horizontal and vertical neighboring cell sub key pool, and 10% with each diagonal neighboring cell sub key-pool. Thus each key in a sub key-pool may be selected by $2N_c$ sensors. Each sensor selects m random keys. Table 3 shows that ETE has adequate connectivity, but the local connectivity with ETE is not as good as LPBK [12] and Yu [16]. However, as will be shown, considering memory usage, computational overhead and resilience against key exposure, ETE is much better than these schemes.

Table 2. The Local Connectivity of Different Techniques

Technique	Local Connectivity
LAKE [14]	0.3228
[13] S=4000, m=50	0.2004
[13] S=4000, m=100	0.4762
[13] S=2000, m=50	0.3286
[13] S=2000, m=100	0.6299
[15] (m=200, t=20, Sc=30)	0.5451
[15] (m=200, t=50, Sc=30)	0.1853
[15] (m=400, t=20, Sc=30)	0. 8883
[15] (m=400, t=50, Sc=30)	0. 4548
ETE	0.8203
Yu et al. [16]	0.9705
LPBK [12]	0.9908

4.3.2 Remote Connectivity

Remote connectivity is the probability of directly producing a common key between two far sensors. This is crucial from the point of view of any two sensors securely communicating with each other in an efficient manner. As mentioned previously, predistributed random keys are used in ETE to establish a common key between far sensors, so we must analyze the probability of key generation in this situation. As described in Section 3.2 the global key pool is divided into N_c sub key-pools. In ETE, two far sensors with the same identity and belonging to different base cells select r random keys from the same sub key-pool. The probability of establishing a common key in this case is

$$
Pr = 1-\binom{Sc}{Sc-r}/\binom{Sc}{r} \qquad (13)
$$

With LAKE, each sensor belonging to a cell can directly establish a common key with only one sensor belonging to another cell. Thus only one sensor can be used as an agent sensor between two cells, so LAKE is vulnerable to a node compromise attack. With the approach in [13], since the sub key-pools for far cells have no correlation, the probability of a common key is close to zero. With LPBK, each cell selects a symmetric polynomial independent of the other cells, so the probability of establishing a common key for two far sensors is exactly zero. In Figure 3 the probability of establishing a common key between two far sensors with ETE is shown for different numbers of random keys stored in the sensors.

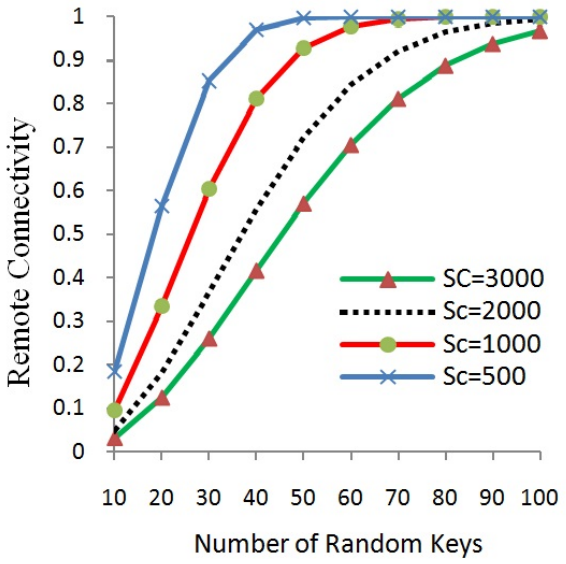

Figure 3. Probability of establishing a common key between two far sensors with ETE

4.3.3 Global Connectivity

Global connectivity is the ratio of the number of nodes in the largest isolated component to the size of the entire network [13]. It is possible that a scheme has high local connectivity but many isolated network components. In this article we want to determine the affect of the key establishment protocol on global connectivity. We assume that every pair of sensors can directly or indirectly reach each other. In other words, we assume that the global connectivity disregarding key establishment is 100%. It is obvious that if the local connectivity is one, the global connectivity will be 100%. To determine the global connectivity, we have selected 1000 sensor nodes randomly and the percentages of sensors who can't establish common key with any other sensors were computed. The results indicate that in the proposed protocol when the local connectivity is 0.50 and the remote connectivity is 0.36, only 0.0008 of the sensors are not accessible via a secure path. Thus global connectivity is not a concern.

4.4 Resilience against Node Compromising Attack

Since the sensor hardware is not tamper proof, an adversary can capture the key material inside it [13]. By compromising some sensors, it may be possible to obtain the common key between uncompromised sensors. In this section, the effect of compromised nodes for both direct and indirect key generation is evaluated for ETE and the other methods.

4.4.1 The Probability of Direct Key Exposure

In methods based on symmetric polynomials, as shown in Section 2.1, when an adversary compromises more than t sensors, they can access the common key between uncompromised sensors. Therefore, in symmetric polynomial based methods, increasing t reduces the probability of revealing uncompromised keys (even to zero). However, a larger t means more processing and memory overhead for sensors, especially in large scale networks. Hence, t should be chosen based on the trade-off between memory and processing costs, and an acceptable security level. As mentioned in Section 3.1.2, ETE restricts key generation with symmetric polynomials to sensors in the same group. If X is the number of compromised sensors in the network and N_i is the number of sensors in each group, an adversary can obtain the common key between two uncompromised sensors if they can compromise at least t sensors in the group. The probability of this event is

$$P_c = \sum_{i=t+1}^{N_i} \frac{\binom{N_i}{i}\binom{N-N_i}{X-i}}{\binom{N}{X}} \tag{14}$$

This expression can be used to evaluate the resilience of ETE, LPBK and LAKE for the corresponding values of N_i, which were given in Section 4.1. With ETE, due to the use of random keys for far sensors, we must also evaluate the effect of compromised sensors on the sensor resilience against random keys being revealed. As mentioned previously, the global key-pool is divided into N_c sub key-pools. Each sensor in a base cell is assigned to a sub key-pool and selects r random keys from it. Since each sensor in a base cell selects keys from a separate sub key-pool, the number of sensors which select the same sub key-pool is N/N_c . If X sensors are compromised, the probability of compromising a sensor will be X/N, and the number of compromised sensors which select the same sub key-pool will be X/N_c . Therefore, the probability of compromising a common key between two uncompromised far sensors is

$$P_c = 1 - \left(1 - \frac{r}{S/N_c}\right)^{\frac{X}{N_c}} \tag{15}$$

In [13], each sensor selects m_R keys from a sub key pool. An adversary can get more information about the sub key pool by compromising more sensors. In this scheme, each cell has a sub key-pool with S keys, and each key exists in two neighboring sub key-pools. Thus, the probability of compromising the key between two sensors that have a common key is

$$1 - \left(1 - \frac{m_R}{S}\right)^{X_i} \tag{16}$$

where X_i is the number of compromised sensors in a group. Since there are two cells in a group, the

probability of compromising X_G sensors in a group when X sensors are compromised in the entire network is

$$P(X_G = i) = \binom{X}{i}(\frac{2}{n})^i(1-\frac{2}{n})^{X-i} \quad (17)$$

where n is the number of cells in the network. Therefore the probability of compromising a common key based on the pre-distributed random key approach in [13] is

$$P_{Com} = 1 - \sum_{i=1}^{min(2N_c,X)} (1-\frac{m_R}{S})^i \binom{X}{i}(\frac{2}{n})^i(1-\frac{2}{n})^{X-i} \quad (18)$$

In [15], each sensor selects key spaces from available key spaces, and each key space is a Blom matrix. Similar to [13], each key space is shared between two neighboring cells and may be selected by $2N_c$ sensors. When t sensors having distinct rows of the same key space are compromised, the other common keys in the key space are also compromised. Therefore, assuming X_i sensors in a group are compromised, the probability of compromising a common key between uncompromised sensors is

$$\sum_{i=t+1}^{X_i} \binom{X_i}{i}(\frac{\tau}{Sc})^i(1-\frac{\tau}{Sc})^{X_i-i} \quad (19)$$

Then the probability of compromising the common key between two uncompromised sensors when X sensors are compromised in the entire network is

$$P_{Com} = \sum_{X_i}^{X} \sum_{i=t+1}^{X_i} \binom{X}{X_i}(\frac{2}{n})^{X_i}(1-\frac{2}{n})^{X-X_i} \binom{X_i}{i}(\frac{\tau}{Sc})^i(1-\frac{\tau}{Sc})^{X_i-i} \quad (20)$$

Based on Equations 14, 18 and 20, the probability of direct key exposure between two uncompromised sensors is shown in Figure 4 for different protocols. It can be seen that for near sensors, ETE provides greater resistance against direct key exposure than the other schemes. When the memory size is greater than 200, the security of LAKE is perfect. In Figure 4, the memory size specifies the value of t for ETE, Du [13], Du [15], LPBK, and Yu [16]. Increasing the memory size will improve the security of these schemes based on Equation 14. As shown in Figure 4, the security of ETE is perfect when the memory size is greater than 400.

4.4.2 The Probability of Indirect Key Exposure

To produce a common key between two sensors not in the same group, agent sensors must be used. Therefore, if agents are compromised, an adversary can access common keys. In [8], it was shown that the average key path length for two adjacent sensors is three sensors, while the length for far sensors is 11 sensors for a network similar to our configuration. The probability of revealing an indirect key, P_{R_I} , between two sensors which use h sensors as agents is

$$P_{R_I} = 1 - \frac{\binom{N-h}{X}}{\binom{N}{X}} \quad (21)$$

We simulated a sensor network with the configuration given in Section 4.1. Our results indicated that with ETE, the average number of agents required for key establishment between far sensors is 1.9821. With LAKE, the corresponding number of agents is 1.3245. In Table 4, the probability of indirect key exposure for different schemes is shown. In this table, the number of agents required for far sensors with LAKE and ETE is assumed to be two. The results in Table 4 show that LAKE and ETE are much more resilient against indirect key exposure than the other schemes.

Table 3. Indirect Key Exposure Probability of Different Schemes

Technique	X=500	X =1000	X=1500	X=2000
LAKE and ETE	0.097	0.19	0.27	0.36
Other Schemes	0.43	0.68	0.83	0.91

4.5 Memory Usage

In this section the memory usage of different schemes is calculated considering perfect security. To achieve perfect security when symmetric polynomials are used, the polynomial degree, t, must satisfy inequality 4. Blom scheme is a t-secure scheme, so that if no more than t nodes are compromised, the link between uncompromised nodes will remain secure. In this case, t specifies the size of the Blom matrix. In ETE, each sensor has $q+1$ polynomial shares from t-degree bivariate symmetric polynomials and r random keys. Since each symmetric polynomial must provide $q+1$ polynomial shares, t must be at least $q-1$. In LPBK, each sensor has five polynomial shares from distinct symmetric polynomials. Each symmetric polynomial must provide $5N_c$ shares, therefore t for each symmetric polynomial must be at least $5N_c-2$. In LAKE, one symmetric polynomial provides polynomial shares for all sensors. Therefore in this scheme t must be at least than N_c-2. With the approach in [16], which is based on Blom scheme, each base cell has six neighboring normal cells, and each normal cell is the neighbor of two base cells. Each normal cell along with its six neighboring cells has a Blom matrix. Therefore each sensor has two rows from two distinct Blom matrices, and the Blom matrix must have size greater than $7N_c$. The memory usage for different schemes is shown in

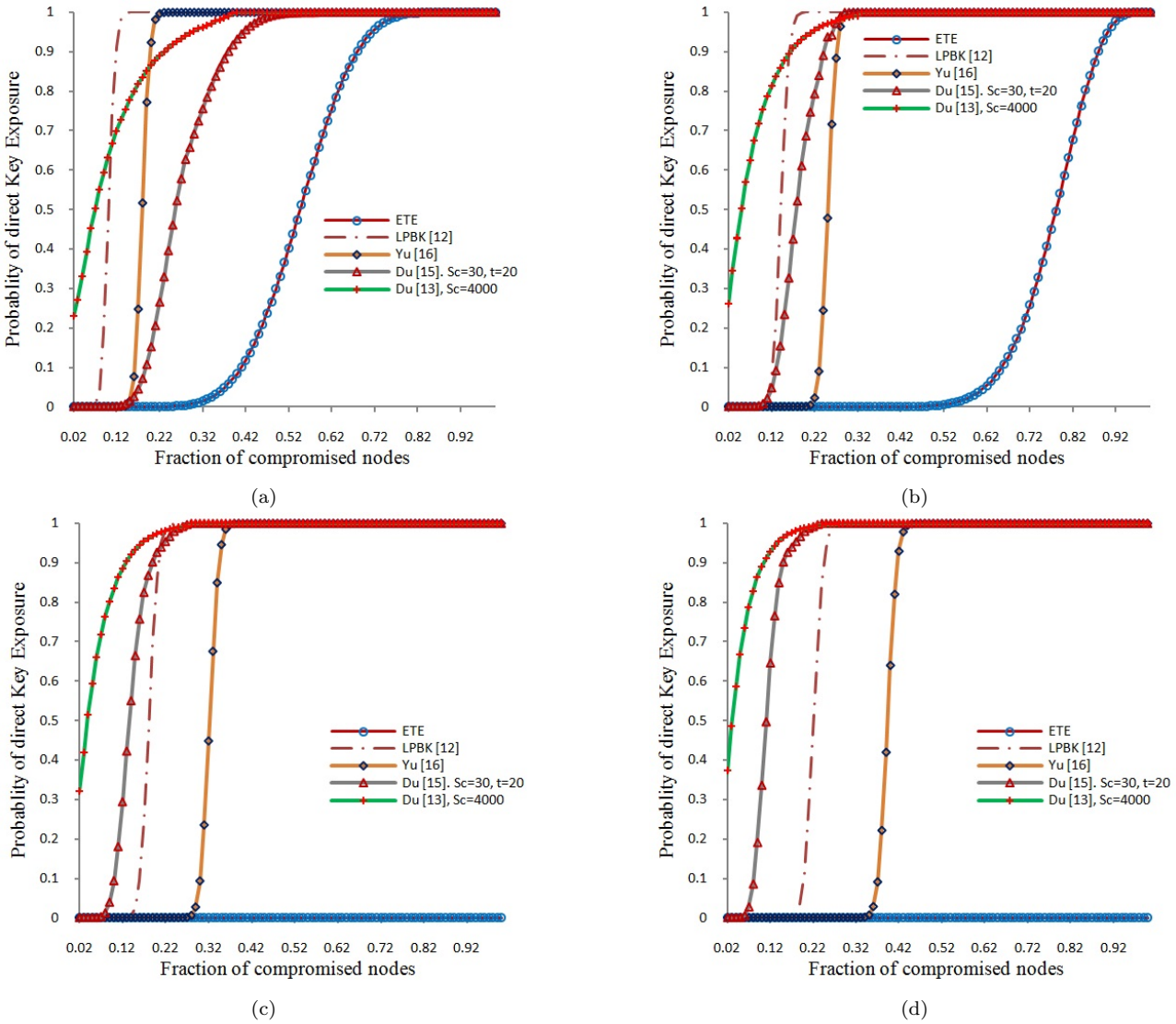

Figure 4. Key resilience with different protocols for available memory (a) 250, (b) 350, (c) 450 and (d) 550.

Table 5. This shows that the memory required with LPBK and the approach in [16] is very large. The memory usage of the technique in [13] depends on the number of random keys (m). However, it cannot provide perfect security.

4.6 Computational Overhead

In sensors, a low power micro-controller is typically used as a processing unit, so we assume that an 8-bit micro-controller from the 8051 family is employed. To compare our proposed protocol with other schemes, we assume sufficient memory is available in the sensor. LPBK, LAKE and our proposed protocol are based on symmetric polynomials. If the polynomial has degree t, $2t$ modular multiplications and t summations are required. We assume a key length of 128 bits. To compute the modular multiplication of two 128 bit numbers, we use the interleaved modulo multiplication algorithm [29] which is implemented as follows

Algorithm 1

Input: X, Y, M with $0 \leq X, Y \leq M$
Output: $P = X \times Y \mod M$
  $n \Leftarrow$ number of bits in X
  $x_i \Leftarrow i^{th}$ bit of X
  $P \Leftarrow 0$
  for $i := n-1$ **to** $i \geq 0$ **step** -1 **do**
    $P \Leftarrow 2 \times P$
    $I \Leftarrow x_i \times Y$
    $P \Leftarrow P + I$
    if $P \geq M$ **then**
      $P \Leftarrow P - M$
    end if
  end for

Our evaluation shows that the number of instruction cycles require to compute the modular multipli-

Table 4. Memory Cost for Different Techniques

Technique	Memory Cost	Required Memory Cost for Direct Key Secrecy
LPBK[12]	$5(t+1)$	$5(5N_c-2)$
Du et al. [15]	$O(\tau(t+1))$	$O(\tau(t+1))$
LAKE [14]	t	$2N_c-2$
Yu et al. [16]	$2(t+1)$	$2(7N_c-2)$
ETE	$(q+1)t+r$	$r+q^2-1, q^2+q+1 \geq 4N_c, 1 \leq r \leq 20$

Table 5. Computational Overhead for Different Schemes

Technique	Threshold for PSC	Average Number of Instruction Cycles per Sensor
ETE	$q-1$	1.58×10^5
LAKE [14]	$2N_c-2$	1.66×10^6
LPBK [12]	$5N_c-2$	4.13×10^6
Yu et al. [16]	$7N_c-2$	5.73×10^6

cation of two 128 bit numbers is approximately 4096 cycles. Hence, the number of instruction cycles is approximately $8300t$ for LPBK, LAKE and our proposed protocol. The value of t in these schemes is not identical and is dependent on the group size. With the approach in [22], which uses Blom scheme for generating common keys between sensors, multiplication of two matrices with dimensions $(1, t+1)$ and $(t+1, 1)$ is required. For this operation, $t+1$ modular multiplications and $t+1$ summations are needed. Therefore, this scheme requires approximately $4096t$ instruction cycles. These results are summarized in Table 5 for the Perfect Security Condition (PSC). This shows that our approach requires far fewer instruction cycles. Moreover, as shown in Figure 3, because the probability of the key establishment between far sensors in ETE, is close to one and the average number of agents required for key establishment between far sensors is 1.9821, as discussed in section 4.4, ETE is more efficient than previous ones in terms of communication and computation overhead.

5 Conclusion

In this article, we introduced a new key management protocol called ETE which is based on symmetric polynomials and random key pre-distribution. Traditional key establishment protocols based on symmetric polynomial use a polynomial to generate polynomial shares for a group of sensors. Due to the large number of sensors in a group, the required polynomial degree is quite high. This imposes a significant cost in terms of memory usage and processing. In the proposed protocol, we use several symmetric polynomials for a group instead of one, and a combinatorial design is used in the distribution of polynomial shares. Further, key establishment between near sensors is provided via symmetric polynomials, while for far sensors key establishment is accomplished via random key pre-distribution. Using both techniques, every pair of nodes can produce a common key either directly or indirectly. The simulation results indicate that the local connectivity with ETE is comparable to LPBK [12] and Yu [16], but the memory usage, computational overhead and resistance against key exposure is very efficient than these schemes. LAKE and ETE are only two schemes that can establish common key between far sensors. However, the local connectivity with ETE is more efficient than LAKE [14].

References

[1] Gregory J. Pottie and William J. Kaiser. Wireless integrated network sensors. *Commun. ACM*, 43 (5):51–58, 2000.

[2] Marcos A. Simplício Jr., Paulo S. L. M. Barreto, Cintia B. Margi, and Tereza Cristina M. B. Carvalho. A survey on key management mechanisms for distributed wireless sensor networks. *Computer Networks*, 54(15):2591–2612, 2010.

[3] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008.

[4] H. T. Kung and Dario Vlah. Efficient location

tracking using sensor networks. In *WCNC*, pages 1954–1961. IEEE, 2003. ISBN 0-7803-7700-1.

[5] R.R. Brooks, P. Ramanathan, and A.M. Sayeed. Distributed target classification and tracking in sensor networks. *Proceedings of the IEEE*, 91(8): 1163–1171, 2003. ISSN 0018-9219. doi: 10.1109/JPROC.2003.814923.

[6] Ronald J. Watro, Derrick Kong, Sue fen Cuti, Charles Gardiner, Charles Lynn, and Peter Kruus. Tinypk: securing sensor networks with public key technology. In Sanjeev Setia and Vipin Swarup, editors, *SASN*, pages 59–64. ACM, 2004. ISBN 1-58113-972-1.

[7] David J. Malan, Matt Welsh, and Michael D. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In *SECON*, pages 71–80. IEEE, 2004. ISBN 0-7803-8796-1.

[8] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In Vijayalakshmi Atluri, editor, *ACM Conference on Computer and Communications Security*, pages 41–47. ACM, 2002. ISBN 1-58113-612-9.

[9] Mi Wen, Yanfei Zheng, Wen jun Ye, Kefei Chen, and Weidong Qiu. A key management protocol with robust continuity for sensor networks. *Computer Standards & Interfaces*, 31(4):642–647, 2009.

[10] Donggang Liu, Peng Ning, and Rongfang Li. Establishing pairwise keys in distributed sensor networks. *ACM Trans. Inf. Syst. Secur.*, 8(1):41–77, 2005.

[11] Yun Zhou and Yuguang Fang. A scalable key agreement scheme for large scale networks. In *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*, pages 631–636, 2006. doi: 10.1109/ICNSC.2006.1673219.

[12] Donggang Liu and Peng Ning. Location-based pairwise key establishments for static sensor networks. In *In 2003 ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN 03*, pages 72–82. ACM Press, 2003.

[13] Wenliang Du, Jing Deng, Yunghsiang S. Han, Shigang Chen, and Pramod K. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In *INFOCOM*, 2004.

[14] Yun Zhou and Yuguang Fang. A two-layer key establishment scheme for wireless sensor networks. *IEEE Trans. Mob. Comput.*, 6(9):1009–1020, 2007.

[15] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney. A key predistribution scheme for sensor networks using deployment knowledge. *IEEE Trans. Dependable Sec. Comput.*, 3(1):62–77, 2006.

[16] Zhen Yu and Yong Guan. A key management scheme using deployment knowledge for wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 19(10):1411–1425, 2008.

[17] A. Fanian and M. Berenjkoub. An efficient end to end key establishment protocol for wireless sensor networks. In *Information Security and Cryptology (ISCISC), 2012 9th International ISC Conference on*, pages 73–79, 2012. doi: 10.1109/ISCISC.2012.6408194.

[18] P Borwein and T Erdélyi. *Polynomials and Polynomial Inequalities*. Springer, New York, 1995.

[19] Yun Zhou and Yuguang Fang. Scalable link-layer key agreement in sensor networks. In *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pages 1–6, 2006. doi: 10.1109/MILCOM.2006.302079.

[20] Ian Anderson. *Combinatorial designs and configurations*. Ellis Horwood Chichester, 1990.

[21] Douglas R. Stinson. *Combinatorial Designs: Constructions and Analysis*. SpringerVerlag, 2004.

[22] Rolf Blom. An optimal class of symmetric key generation systems. In Thomas Beth, Norbert Cot, and Ingemar Ingemarsson, editors, *EUROCRYPT*, volume 209 of *Lecture Notes in Computer Science*, pages 335–338. Springer, 1984. ISBN 3-540-16076-0.

[23] Ali Fanian, Mehdi Berenjkoub, Hossein Saidi, and T. Aaron Gulliver. A new key establishment protocol for limited resource wireless sensor networks. In *CNSR*, pages 138–145. IEEE Computer Society, 2010.

[24] T. Kavitha and D.Sridharan. Hybrid design of scalable key distribution for wireless sensor networks. *IACSIT International Journal of Engineering and Technology*, 2(2):136–141, 2010.

[25] Ali Fanian, Mehdi Berenjkoub, Hossein Saidi, and T. Aaron Gulliver. An efficient symmetric polynomial-based key establishment protocol for wireless sensor networks. *ISeCure, The ISC International Journal of Information Security*, 2(2): 89–105, 2010.

[26] Chris Karlof and David Wagner. Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Networks*, 1(2-3):293–315, 2003.

[27] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wireless Networks*, 11 (1-2):21–38, 2005.

[28] Gergely Ács, Levente Buttyán, and István Vajda. Provably secure on-demand source routing in mobile ad hoc networks. *IEEE Trans. Mob. Comput.*, 5(11):1533–1546, 2006.

[29] Viktor Bunimov and Manfred Schimmler. Area and time efficient modular multiplication of large integers. In *ASAP*, pages 400–. IEEE Computer Society, 2003. ISBN 0-7695-1992-X.

Ali Fanian received the Ph.D. degree from the Department of Electrical and Computer Engineering, Isfahan University of Technology in 2011. The title of his dissertation is Key Management Protocol in Wireless Sensor Networks. He started work in the same department as an Assistant Professor at that time. His research interests include ad-hoc networks, wireless network security and hardware design.

Mehdi Berenjkoub received the Ph.D. degree from Department of Electrical and Computer Engineering, Isfahan University of Technology in 2000. The title of his dissertation is two-party key distribution protocols in cryptography. He started his work in the same department as an assistant professor from that time. Graduate courses presented by him include Fundamentals of Cryptography, Cryptographic Protocols, Network Security, and Intrusion Detection. He has supervised more than a dozen M.Sc. students and Ph.D. candidates in related areas. He also was one of the founder members for Iranian Society of Cryptology in 2001. He has continued his cooperation with the society as an active member. He along with his colleagues recently established a research group on Security in Networks and Systems in IUT. He also is responsible for a newly established academic CSIRT in IUT. His current interested research topics are wireless network security and authentication protocols.