Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 216754, 13 pages
http://dx.doi.org/10.1155/2013/216754

Research Article

An Energy-Efficient Key Predistribution Scheme for Secure Wireless Sensor Networks Using Eigenvector

Sung Jin Choi, Kyung Tae Kim, and Hee Yong Youn

College of Information and Communication Engineering, Sungkyunkwan University, Suwon 440-746, Republic of Korea

Correspondence should be addressed to Hee Yong Youn; youn@ece.skku.ac.kr

Received 4 January 2013; Revised 5 May 2013; Accepted 8 May 2013

Academic Editor: Laurence T. Yang

Currently, sensor networks are widely used in various fields. Here secure operations are required for critical applications since the damages are significant if the network is compromised or disrupted. For the security of wireless sensor network, the earlier schemes typically employ asymmetric cryptography. These schemes are, however, often unsuitable for wireless sensor network due to the limited computational power and energy of the sensor nodes. To address this issue, various approaches have been developed, and the random key predistribution approach has been recognized as an effective approach. One shortcoming, however, is that a common key is not guaranteed to be found between any two nodes wanting to communicate. This paper proposes a new robust key predistribution scheme solving this problem, with which the security is not compromised even though the data exchanged between the nodes are tapped by an adversary. This is achieved by using the keys assigned based on the notion of eigenvalue and eigenvector of a square matrix of a pool of keys. Mathematical analysis and computer simulation reveal that the proposed scheme significantly reduces the overhead required for secure connectivity and energy consumption of sensor nodes compared to the existing approaches.

1. Introduction

Wide-spread deployment of sensor networks is quite practical these days. A network of thousands or more sensors allows an efficient solution to various challenging tasks: traffic monitoring, monitoring of building with respect to the structure, fire, and security, military sensing and tracking, distributed measurement of seismic activity, real-time pollution monitoring, wild life monitoring, wild fire tracking, and so forth [1–3]. Energy-aware distributed intelligent data gathering with wireless sensor networks is a hot issue lately due to the emerge of big data paradigm [4].

Wireless sensor networks (WSNs) share several common properties with the traditional wireless networks. Both of them include arrays of nodes that are battery powered, have limited computational capabilities and memory, and rely on intermittent wireless communication via radio frequency and, possibly, optical links. They also include *data-collecting nodes* which cache the sensed data and make them available to the processing components of the network and *control nodes* which monitor the status of the sensor nodes and broadcast simple commands to them. However, WSNs differ from the traditional wireless networks in several aspects; namely, the scale is a few orders of magnitude larger than that of wireless networks; they are dynamic in the sense that addition and removal of sensor nodes are allowed after the deployment to expand the network or replace failed or unreliable nodes without physical contact; and they may be deployed in hostile areas where the communication is monitored and the sensor nodes are subject to capture and manipulation by an adversary. These harsh operational conditions place very critical security constraints on the WSN design [5–9].

Numerous commercial and military applications require secure operation of sensor networks, and seriously detrimental outcomes might be caused if the network is compromised or disrupted. When the sensor networks are deployed in a hostile environment, security is extremely important as they are prone to different types of malicious attacks. For example, an enemy can easily tap the information, imitate one of the sensor nodes, or intentionally provide incorrect information to other nodes. The critical issue here is how to secure the communication between the sensor nodes; that is, how to set

up a secret key between the communicating nodes. Most of the earlier schemes use asymmetric cryptography to solve this problem [10]. However, these schemes are often unsuitable to distributed sensor network due to limited computational power and energy of sensor nodes.

To address this issue a scheme has been proposed, which is based on random key pre-distribution. However, it has a shortcoming that a common key is not guaranteed to be found between any two nodes wanting to communicate, and there is also a high possibility of leakage of key information and breakdown of security. This is because the keys are distributed using an identifier, working as a key transport between the sensor nodes. This paper proposes a new key pre-distribution scheme solving this problem by assigning the keys based on the notion of eigenvalues and eigenvectors [11] of a square matrix of a pool of random keys. The main idea here is that there exists infinite combination of eigenvalues and eigenvectors building a matrix. As a result, one cannot ever conjecture the original matrix with only a portion of eigenvalues and eigenvectors of the sensor nodes, which corresponds to the shared key. It thus provides high security to wireless sensor networks of pre-distributed keys by not exposing any data on the key to other nodes. The main advantages and contributions are summarized as follows.

(i) A common key is guaranteed to be found between any two nodes wanting to communicate.

(ii) The key cannot be leaked unless entire sensor nodes are compromised.

(iii) It requires much smaller memory space to hold the pre-distributed keys.

(iv) The energy efficiency is higher.

Analytical modeling and computer simulation reveal that the proposed scheme significantly reduces the overhead required for secure connectivity and energy consumption of the sensor nodes compared to the existing approach employing the random key pre-distribution scheme.

The rest of the paper is organized as follows. Section 2 discusses the existing key distribution approaches for sensor networks, and Section 3 presents the proposed scheme. Section 4 analyzes and compares the performance of the proposed scheme with those of the earlier schemes, and finally concluding remarks are given in Section 5.

2. Related Works

There exist a number of key pre-distribution schemes developed for wireless sensor network. A basic approach is to let all the nodes carry a master secret key, and any pair of nodes use the global master key for key agreement and creation of a new pairwise key. This approach does not exhibit sufficient network resilience such that if one node is compromised, the security of the entire sensor network is compromised. Some existing studies suggest storing the master key in tamper-resistant hardware to reduce the risk [12], but this increases the cost and energy consumption of each sensor node. Furthermore, tamper-resistant hardware might not always be safe. Liu and Ning [13] proposed another key pre-distribution scheme which substantially improves the resilience of the network compared to other schemes. This scheme exhibits a threshold property; when the number of compromised nodes is smaller than the threshold, the probability that other noncompromised nodes are affected is close to zero. This desired property lowers the initial payoff of small-scale network breaches to an adversary, and makes it necessary for the adversary to attack a significant portion of the network.

Blundo et al. [14] proposed several schemes which allow any group of some parties to compute a common key while being secure against collusion between some members of them. These schemes focus on saving communication cost while memory constraints are not placed on the group members. Perrig et al. [10] proposed SPINS, a security architecture specifically designed for sensor networks. In SPINS, each sensor node shares a secret key with the base station, and any pair of sensor nodes cannot directly establish a secret key. They use the base station as a trusted third party to set up a secret key.

Eschenauer and Gligor [15] proposed a random key pre-distribution scheme, which exploits the probabilistic characteristics of random graph. In this scheme the basestation first creates a large number of random keys and saves them in the key pool. Then, a group of keys are randomly selected from it to build a key ring, which is distributed to the sensor nodes. The sensor nodes find the shared keys among the neighboring nodes residing within the wireless communication radius by broadcasting the key ring and key information to the neighbor nodes. Any two nodes apart by two or more links or having no shared key have to create a path key in order to have a shared key. One of the two nodes selects a key from the key ring and transmits it to other nodes through the intermediate nodes in the key path until reaching the target node. The operation of this scheme consists of three phases as follows, which is illustrated in Figure 1.

2.1. Phase I (The Initialization). The Eschenauer and Gligor (E-G) scheme randomly decides a pool of keys, P, out of the key space generated by random graph. In each node k keys are randomly selected from P and stored in the memory. This set of k keys is called the node's key ring. The number of keys in the key pool, $|P|$, is chosen such that two random subsets of size m in P will share at least one key with some probability p.

2.2. Phase II (The Discovery of Shared Key). The nodes perform key discovery to find out shared key from their neighbors. The key discovery is performed by assigning a short identifier to each key prior to deployment and having each node broadcast its set of identifiers. The nodes identifying that they contain a shared key in their key ring can then verify that their neighbor actually holds the key through a challenge response protocol. The shared key then becomes the key for that link.

2.3. Phase III (The Setup of Path Key). The nodes can set up path key with the nodes in their vicinity when they cannot

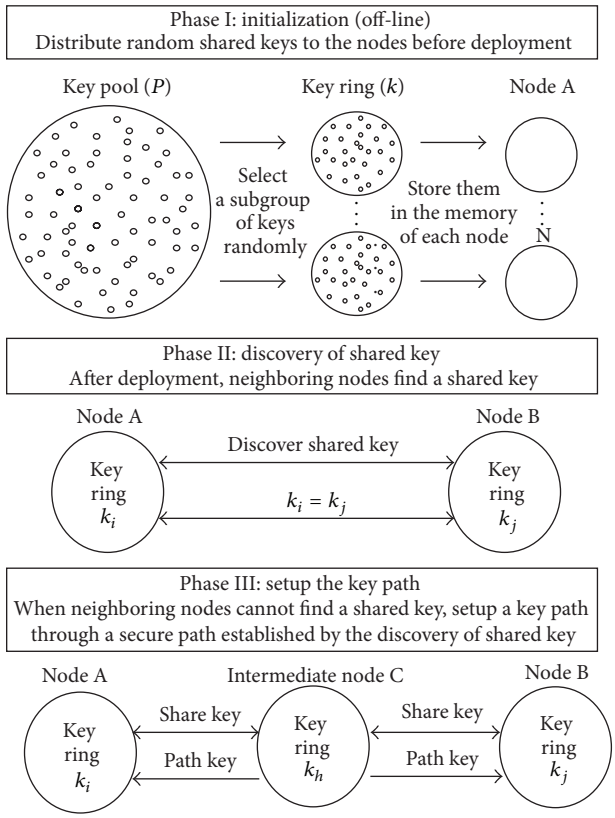

Figure 1: The procedure of the Eschenauer and Gligor scheme.

find shared keys from their key rings. If the network is connected, a path can be found from a source node to its neighbor. The source node then generates a path key and sends it securely via the path to the target node.

The key pre-distribution scheme proposed by Chan et al. [16] also employs the random graph approach like the scheme proposed in [15], but it uses q (≥ 1) shared keys instead of one. This scheme connects two nodes via multiple paths and creates the keys to fortify the security. As a result, even if a sensor node is damaged by the attacker, the security of the rest of the nodes can be preserved using the random/shared keys. This scheme decides a pool of keys of size of P from the key space and composes a key ring of k elements. For the communication of one node with the neighbor nodes, at least more than q keys need to be shared and security is provided by creating a new key ($\text{hash}(k_1|k_2|\cdots k_q)$). This scheme focuses on the security fortification against small-scale attacks. One shortcoming here is that a shared key between any two nodes is not guaranteed to be found. Moreover, it does not support the mechanism for mutual authentication between the nodes.

Camtepe and Yener [17] and Lee and Stinson [18] applied combinatorial approaches to key pre-distribution. They presented two classes of combinatorial designs: symmetric balanced incomplete block designs and generalized quadrangles. The points and blocks in the combinatorial designs are associated with distinct key identifiers and nodes, respectively. Here even though the probability of key establishment has been increased, the network resiliency is still limited and node authentication is not ensured. Sánchez and Baldus [19] made use of combinatorial design theory for the pre-distribution of multiple bivariate polynomial shares based on [14]. Their approach enables direct pairwise key establishment for a large number of nodes, independently of the physical connectivity properties of WSNs. Chakrabarti et al. [20] also used the combinatorial designs for key pre-distribution in WSNs. Their method is to begin with a transversal design and then form the key rings by merging the blocks. Some performance metrics are improved at the cost of larger storage.

Liu et al. [21] proposed an asymmetric key pre-distribution scheme (AKPS). AKPS uses a trusted authority (TA) to distribute secret keys to each user and public keying material to keying material servers (KMSs). With the help of KMSs, two sensor nodes can establish a session key to encrypt messages. AKPS has an advantage over other schemes in that the compromise of KMSs does not disclose any information of the users' secret keys and the session keys. Nguyen et al. proposed a key management scheme considering the signal range in [22]. Each node is assigned with a subset of keys from the key pool by a key setup server. Two nodes residing in each other's communication range is assigned a subset of common keys. This scheme also includes shared key discovery and path key establishment phases. By using the location information of sensor nodes, it improves the connectivity and achieves better resilience than other schemes. However, this scheme depends on the information of sensor deployment.

Szczechowiak and Collier [23] proposed a key agreement scheme based on identity-based cryptograph (IBC) for wireless sensor networks. A trust authority is used to predistribute a secret key, a unique identity IDx, a hashing function H, a mapping function, and a key derivation function (KDF) into the memory of each node. This scheme saves much key storage space and allows high resilience against node capture. However, the key agreement protocol employs pairing-based cryptography, which requires large computational and energy resource for each sensor node to compute the shared pairwise keys together with its neighboring nodes.

Some literatures focus on localizing the keys. In [24, 25] the authors presented RPKH and location-dependent key management (LDK) scheme to allow local key management. They utilize different nodes including the normal nodes and anchor nodes to generate the keys of different transmission ranges. The LDK scheme employs heterogeneous sensors to build a clustered sensor network; the higher-ability nodes (anchor nodes) take the management role and regular nodes. The anchor nodes use the location information of other nodes to generate sets of keys. The neighboring nodes establish secure communication link by determining common keys via exchanging the data of their key. LDK takes advantage of relative location of the nodes by utilizing the anchor nodes of different power level. According to the locations, the nodes receive different sets of keys from the anchor nodes, and the neighboring nodes can establish secure communication link through the common keys. LDK can increase the direct connectivity ratio among the nodes. However, the nodes need to transmit a message containing all the data of the key for determining common keys. This operation consumes lots of

energy, and thus it is not appropriate for WSNs. Moreover, the adversary can eavesdrop on the exchanged key data, and the anchor nodes are difficult to deploy.

Recently, efficient and secure key management for wireless sensor networks attracted a number of researchers [26–28]. Bechkit et al. [29] and Gu et al. [30] focused on key pre-distribution approach for mainly scalability, and Kim et al. [31] applied the key distribution to the clustered WSN. A new polynomial-based rekeying scheme was also proposed by Guo and Qian [32], and an adaptive dynamic key management approach was proposed by Alcaraz et al. [33]. As a different approach, Yu and Wang [34] and Paterson and Stinson [35] suggested to use combinatorial design. Salam et al. [36] proposed public key cryptography for key pre-distribution. An asymmetric matrix and projective plane was used by Subash and Divya [37] and Mitra et al. [38], respectively. The distinctive features of the proposed scheme compared to these key pre-distribution approaches are that it takes advantages of the notion of eigenvectors of a symmetric matrix, which disallows reverse mapping (and thus compromise of security).

The two main issues in the random key pre-distribution approaches are guaranteeing to find a common key between any pair of nodes and the prevention of information leaks. We next present the proposed scheme effectively handling these issues.

3. The Proposed Scheme

In this section the proposed scheme is presented, deferring the analysis and performance evaluation on the security and energy efficiency to the next section. The proposed key pre-distribution scheme employs the random graph approach like Eschenauer's method [15]. However, it guarantees that any pair of nodes can find the shared key between them while preventing the leakage of key information.

3.1. Preliminaries. The proposed scheme is based on important properties of a matrix in designing the key pre-distribution scheme.

Definition 1 (eigenvalue and eigenvector). Let A be an $n \times n$ matrix. A nonzero vector v is an eigenvector of A if (1) holds for some scalar λ. λ is called an eigenvalue of A corresponding to the eigenvector v. Eigenvalues are also known as characteristic, or proper, values or even as latent roots

$$Av = \lambda v. \tag{1}$$

Let us now discuss how to compute eigenvalues and eigenvectors in general. Because $Av = \lambda v \Rightarrow Av = \lambda Iv, \Rightarrow Av - \lambda Iv = 0 \Rightarrow (A - \lambda I)v = 0$, we see that v is an eigenvector, if and only if it is a nontrivial solution of the homogeneous system $(A - \lambda I)v = 0$. In this case, v is a nonzero vector of the null space of $A - \lambda I$. The system has a nontrivial solution, if and only if the determinant of the coefficient matrix is zero. Thus, λ is an eigenvalue of A, if and only if $\det(A - \lambda I) = 0$ [39].

Definition 2 (α-*secure*). As long as an adversary compromises less than or equal to α nodes, uncompromised nodes are perfectly secure.

The security of the proposed scheme is stronger than α–secure in the sense that the entire security is unbroken despite $(\alpha + 1)$ nodes being exposed. The entire security could be broken only when the entire keys pre-distributed are leaked, which is virtually impossible.

3.2. The Proposed Key Distribution Scheme. The key pre-distribution scheme proposed in this section randomly selects k keys out of the key pool of p elements and then generates the index of these keys. A random function is used to generate node identifiers, and the keys generated in the key pool are used as session keys.

The session key is an encryption key used for only one communication session. In case a key is used for numerous encryption messages, the key could be extracted from the messages. This is prevented using a temporary session (i.e., one-time) key. The session key approach employed in the earlier schemes may cause key exposure when used repeatedly. This problem is solved by the proposed approach using the pre-distributed key combination (initial vector).

3.2.1. Setup of Initial Vector. The initial vector is set via four off-line steps: (i) generation of a large pool of keys (e.g., $2^{17} \sim 2^{20}$ keys), (ii) formation of a square matrix using the pool of keys, (iii) derivation of eigenvalues and eigenvectors for the square matrix, (iv) and key pre-distribution to each sensor node.

Step 1 (generation of a large pool of keys). The proposed key pre-distribution scheme is based on random keys. Therefore, a large pool of keys (e.g., $2^{17} \sim 2^{20}$ keys) are generated in this step. Each sensor node receives a subset of keys from the pool before deployment. For the communication between two nodes, they need to find one common key to be used as a shared secret key.

Step 2 (forming a square matrix using the pool of keys). Eschenauer's scheme uses just a pool of keys. However, the proposed scheme uses a pool of keys formed in a square matrix. The random keys are first laid out in the square matrix format before applying the proposed key pre-distribution scheme using eigenvalues and eigenvectors.

Step 3 (deriving eigenvalues and eigenvectors for the square matrix). The eigenvalues and eigenvectors derived from the square matrix are stored as keys in each sensor node. It is to allow a common key between any two nodes and increase the security by providing node-to-node mutual authentication.

A general method for finding eigenvalues and eigenvectors shown in Definition 1 needs to be developed. It computes the dominant eigenvalue and eigenvector corresponding to the dominant eigenvalue. Without loss of generality, it is necessary to assume that square matrix, A, has the following two properties.

(i) There is a single eigenvalue of maximum modulus.

(ii) There is a linearly independent set of n eigenvectors.

According to the first assumption, the eigenvalues can be labeled such that

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|. \quad (2)$$

According to the second assumption, there is a basis $\{v^{(1)}, v^{(2)}, \ldots, v^{(n)}\}$ for C^n such that

$$Av^{(j)} = \lambda_j v^{(j)} \quad (1 \leq j \leq n). \quad (3)$$

Let $x^{(0)}$ be an element of C^n such that when $x^{(0)}$ is expressed as a linear combination of the basis elements $v^{(1)}, v^{(2)}, \ldots, v^{(n)}$, the coefficient of $v^{(1)}$ is not 0. Thus,

$$x^{(0)} = a_1 v^{(1)} + a_2 v^{(2)} + \cdots + a_n v^{(n)} \quad (a_1 \neq 0). \quad (4)$$

We form then $x^{(1)} = Ax^{(0)}, x^{(2)} = Ax^{(1)}, \ldots, x^{(k)} = Ax^{(k-1)}$ to have

$$x^{(k)} = A^k x^{(0)}. \quad (5)$$

In the following analysis there is no loss of generality in absorbing all the coefficients a_j in the vectors $v^{(j)}$. By (5), we have

$$x^{(k)} = A^k v^{(1)} + A^k v^{(2)} + \cdots + A^k v^{(n)}. \quad (6)$$

Using (3), we arrive at

$$x^{(k)} = \lambda_1^k \left[v^{(1)} + \left(\frac{\lambda_2}{\lambda_1}\right)^k v^{(2)} + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^k v^{(n)} \right]. \quad (7)$$

Since $|\lambda_1| > |\lambda_j|$ for $2 \leq j \leq n$, we see that the coefficients $(\lambda_j/\lambda_1)^k$ tend to 0 and the vector within the brackets converges to $v^{(1)}$ as $k \to \infty$.

To simplify the notation, we write $x^{(k)}$ in the form

$$x^{(k)} = \lambda_1^k \left[v^{(1)} + \varepsilon^{(k)} \right] \quad \text{where } \varepsilon^{(k)} \longrightarrow 0 \text{ as } k \longrightarrow \infty. \quad (8)$$

In order to be able to take ratios, let φ be any linear functional on C^n for which $\varphi(v^{(1)}) \neq 0$. Recall that a linear functional φ satisfies $\varphi(\alpha x + \beta y) = \alpha\varphi(x) + \beta\varphi(y)$, for scalars α and β and vectors x and y. (e.g., φ could simply evaluate the jth component of any given vector). Then

$$\varphi\left(x^{(k)}\right) = \lambda_1^k \left[\varphi\left(v^{(1)}\right) + \varphi\left(\varepsilon^{(k)}\right) \right]. \quad (9)$$

Consequently, the following ratios converge to λ_1 as $k \to \infty$:

$$r_k \equiv \frac{\varphi\left(x^{(k+1)}\right)}{\varphi\left(x^{(k)}\right)} = \lambda_1 \left[\frac{\varphi\left(v^{(1)}\right) + \varphi\left(\varepsilon^{(k+1)}\right)}{\varphi\left(v^{(1)}\right) + \varphi\left(\varepsilon^{(k)}\right)} \right] \longrightarrow \lambda_1. \quad (10)$$

Since the direction of the vector $x^{(k)}$ aligns more and more with $v^{(1)}$ as $k \to \infty$, this results in the eigenvector $v^{(1)}$. If the eigenvectors found are

$$v^{(1)} = \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix} s_n \in R, \qquad v^{(2)} = \begin{bmatrix} t_1 \\ \vdots \\ t_n \end{bmatrix} t_n \in R, \ldots,$$
$$v^{(n)} = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} z_n \in R, \quad (11)$$

P ($P = [v^{(1)} v^{(2)} \cdots v^{(n)}]$) is actually stored key and D matrix consisting of eigenvalues becomes the decryption key. An important property of the proposed scheme is that it allows both key pre-distribution and encryption of data at the same time. That is, one cannot extract the original data even though some elements of the P matrix stored in each sensor node are leaked. This is because deciding the original matrix using a small portion of P matrix is impossible. As a result, the proposed key pre-distribution scheme allows high security.

Step 4 (key pre-distribution). In this step every node is assigned P matrix consisting of eigenvectors and D matrix consisting of eigenvalues. Note here that there exist infinite ways for forming P matrix by arbitrarily deciding the s_n, t_n, and u_n values. In addition, only the diagonal elements (eigenvalues) from the D matrix are stored to minimize the required memory space.

3.2.2. Distribution of Session Key. The previous subsection explained how to generate the initial vector using pre-distributed key combinations. This subsection describes how to distribute the session keys from a source node to the destination node using the initial vector. It consists of four steps: (i) set the initial vector between two nodes, (ii) exchange messages for setting a session, (iii) set the session key, (iv) and update the initial vector.

Step 1 (set the initial vector between two nodes). An initial vector needs to be set between two nodes for which a security session is to be set. This step needs Definition 3.

Definition 3 (eigenvalue and eigenvector). Assume that an $n \times n$ matrix A can be converted to a diagonal matrix D, which is called diagonalizable. Then there exists an invertible $n \times n$ matrix P such that

$$P^{-1}AP = D. \quad (12)$$

The process of finding matrices P and D is called diagonalization. First, it is worth noticing that if D is a diagonal matrix with diagonal entries $\lambda_1, \ldots, \lambda_n$, then for $i = 1, \ldots, n$

$$De_i = \lambda e_i. \quad (13)$$

Hence, the standard basis vectors $e_1, \ldots, e_n$ are eigenvectors of D. In particular, the eigenvectors of D are linearly independent. To find a common key A using Definition 3, the following theorem needs to be proved.

Theorem 4. *One has the following.*

(i) *A is diagonalizable if and only if it has n linearly independent eigenvectors.*

(ii) *If A is diagonalizable with $P^{-1}AP = D$, then the columns of P are eigenvectors of A and the diagonal entries of D are the corresponding eigenvalues.*

(iii) *If $\{v_1, \ldots, v_n\}$ are linearly independent eigenvectors of A with corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$, then A can be diagonalized by*

$$P = \left[v_1 v_2 \cdots v_n\right], \qquad D = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}. \tag{14}$$

Proof. Let P be a matrix with columns of n-vectors $v_1, \ldots, v_n$ and let D be a diagonal matrix with diagonal elements, $\lambda_1, \ldots, \lambda_n$, respectively. Then

$$\begin{aligned} AP &= A\left[v_1 v_2 \cdots v_n\right] = \left[v_1 v_2 \cdots v_n\right] \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \\ &= PD. \end{aligned} \tag{15}$$

If A is diagonalizable, with $P^{-1}AP = D$, then $AP = PD$. Hence, $Av_i = \lambda_i v_i$, $i = 1, \ldots, n$. So, the λ_is are eigenvalues and the v_is are corresponding eigenvectors. □

Suppose that A has n linearly independent eigenvectors, say, $v_1 v_2 \cdots v_n$ (the columns of P). If $\lambda_1, \ldots, \lambda_n$ are the corresponding eigenvalues, then $Av_i = \lambda_i v_i, i = 1, \ldots, n$. If D is a diagonal matrix with diagonal entries $\lambda_1, \ldots, \lambda_n$, then $AP = PD$ by (15). Because P is a square matrix with linearly independent columns, it is invertible. Hence, $P^{-1}AP = D$, and A is diagonalizable [40].

If we multiply P and P^{-1} matrix to both sides of (12), it becomes $PP^{-1}APP^{-1} = PDP^{-1}$. Since $PP^{-1} = I$, $P^{-1}P = I$ (I: identity matrix),

$$\begin{aligned} A &= PDP^{-1} \\ &= \left[v^{(1)} \cdots v^{(n)}\right] \begin{bmatrix} \lambda_1 & \cdots & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & \lambda_n \end{bmatrix} \left[v^{(1)} \cdots v^{(n)}\right]^{-1}. \end{aligned} \tag{16}$$

Therefore, the common key can be found from P and D matrix that was stored in the sensor nodes by $A = PDP^{-1}$.

Assume that node_x and node_y contain $(v_1^1, v_1^2 \ldots, v_1^n)$ and $(v_2^1, v_2^2, \ldots, v_2^n)$, respectively. When node_x and node_y need to find the initial vector between them, they first exchange randomly selected vectors and then compute a vector product as follows;

$$\begin{aligned} \text{node}_x &: [v_1^1 \cdots v_2^k \cdots v_1^n][\lambda_1 \cdots \lambda_n][v_1^1 \cdots v_2^k \cdots v_1^n]^{-1}, \\ \text{node}_y &: [v_2^1 \cdots v_1^k \cdots v_2^n][\lambda_1 \cdots \lambda_n][v_2^1 \cdots v_1^k \cdots v_2^n]^{-1}. \end{aligned}$$

Recall that A is diagonalizable, and thus A is used as an initial vector between node_x and node_y.

Step 2 (exchange messages for setting a session). Using the pre-distributed key combinations as the initial vector, m keys from the k keys are selected at each node. Also, $T_{ck(i)}$ in the source node and $D_{ck(i)}$ in the destination node are generated by applying Exclusive-OR operation to the key selected. Using $T_{ck(i)}, D_{ck(i)}$, and the initial vector $SD_{\text{IV}(k)}$, an encrypted message required for setting a session is exchanged between the nodes. They are MS $(= SD_{\text{IV}(k)} \oplus T_{ck(i)})$ and MD $(= SD_{\text{IV}(k)} \oplus D_{ck(i)})$.

Step 3 (set the session key). In this step, with the messages exchanged in Step 2 used to set a session at each node, the session key is generated. The message exchanged to set the session extracts $T_{ck(i)}$ and $D_{ck(i)}$ with the Exclusive-OR operation, and each node generates $SD_{sk(i)} = MS \oplus MD = T_{ck(i)} \oplus D_{ck(i)}$ as the session key.

Step 4 (update initial vector). A secure communication is available at each node using the session key decided in Step 3. In this step the initial vector is updated to improve the security level at each node. The updated initial vector $SD_{\text{IV}(i+1)}$ is extracted by the messages setting the session, MS and MD, and the initial vector, $SD_{\text{IV}(i)}$, as in the following expression:

$$SD_{\text{IV}(k+1)} = MS \oplus MD \oplus SD_{\text{IV}(k)}. \tag{17}$$

The two nodes now share $SD_{sk(i)}$ as the session key. Figure 2 depicts the flow of message exchange between the two communicating nodes.

3.3. Example

3.3.1. Setup of Initial Vector

Step 1 (generation of a large pool of keys).

Step 2 (forming a square matrix using the pool of keys). We have

$$\begin{bmatrix} 3 & 0 & 0 \\ -4 & 6 & 2 \\ 16 & -15 & -5 \end{bmatrix}. \tag{18}$$

Step 3 (deriving eigenvalues and eigenvectors for the square matrix).

The eigenvalues obtained are $\lambda_1 = 0, \lambda_2 = 1$, and $\lambda_3 = 3$. Eigenvectors corresponding to each λ become

$$v^1 = \begin{bmatrix} 0 \\ s \\ -3s \end{bmatrix}, \qquad v^2 = \begin{bmatrix} 0 \\ 2t \\ -5t \end{bmatrix}, \qquad v^3 = \begin{bmatrix} u \\ 0 \\ 2u \end{bmatrix} \tag{19}$$

$$s, t, u \in R.$$

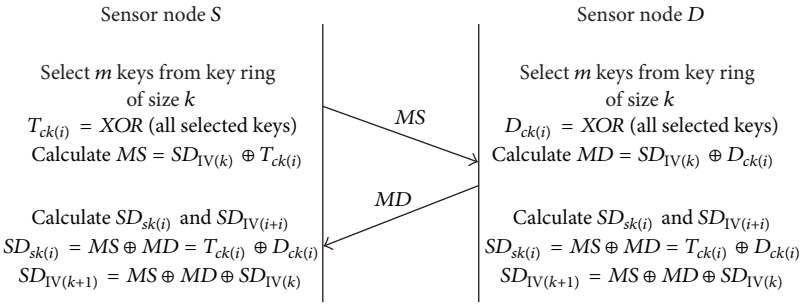

Figure 2: The flow of message exchange between the nodes.

Therefore, the actually stored key P and decryption key D are as follows:

$$P = \left[v^1, v^2, v^3\right], \qquad D = \left[0 \;\; 1 \;\; 3\right]. \tag{20}$$

Step 4 (key pre-distribution). We have

$$\begin{aligned} \text{node}_x : P &= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \\ -3 & -5 & 2 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \\ & \qquad\qquad s = t = u = 1, \\ \text{node}_y : P &= \begin{bmatrix} 0 & 0 & 2 \\ 1 & -2 & 0 \\ -3 & 5 & 4 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \\ & \qquad\qquad s = 1, t = -1, u = 2. \end{aligned} \tag{21}$$

3.3.2. Distribution of Session Key

Step 1 (set the initial vector between two nodes). When node_x and node_y need to find the initial vector between them, they first exchange randomly selected vectors v^3 (v^3 of the node_x is [1 0 2], v^3 of the node_y is [2 0 4]) and then compute a vector product as follows:

$$\begin{aligned} \text{node}_x &: \begin{bmatrix} 0 & 0 & 2 \\ 1 & 2 & 0 \\ -3 & -5 & 4 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 2 \\ 1 & 2 & 0 \\ -3 & -5 & 4 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 3 & 0 & 0 \\ -4 & 6 & 2 \\ 16 & -15 & -5 \end{bmatrix} = SD_{\text{IV}(1)}, \\ \text{node}_y &: \begin{bmatrix} 0 & 0 & 1 \\ 1 & -2 & 0 \\ -3 & 5 & 2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & -2 & 0 \\ -3 & 5 & 2 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 3 & 0 & 0 \\ -4 & 6 & 2 \\ 16 & -15 & -5 \end{bmatrix} = SD_{\text{IV}(1)}. \end{aligned} \tag{22}$$

Step 2 (exchange message for setting a session). We have

$T_{ck(3)} = [1\ 0\ 2]$ (3 rd column vector of the node_x),
$SD_{\text{IV}(1)} = [-15 \;\; -5 \;\; -4 \;\; 0 \;\; 2 \;\; 3 \;\; 6 \;\; 16]$

$MS = SD_{\text{IV}(1)} \oplus T_{ck(3)} = [-15 \;\; -5 \;\; -4 \;\; 0 \;\; 1 \;\; 2 \;\; 3 \;\; 6 \;\; 16]$

$D_{ck(3)} = [2\ 0\ 4]$ (3 rd cloumn vector of the node_y),
$SD_{\text{IV}(1)} = [-15 \;\; -5 \;\; -4 \;\; 0 \;\; 2 \;\; 3 \;\; 6 \;\; 16]$

$MD = SD_{\text{IV}(1)} \oplus D_{ck(3)} = [-15 \;\; -5 \;\; -4 \;\; 0 \;\; 2 \;\; 3 \;\; 4 \;\; 6 \;\; 16]$.

Step 3 (set the session key). $SD_{\text{IV}(1)} = MS \oplus MD = [-15 \;\; -5 \;\; -4 \;\; 0 \;\; 1 \;\; 2 \;\; 3 \;\; 4 \;\; 6 \;\; 16]$.

Step 4 (update initial vector). $SD_{\text{IV}(2)} = MS \oplus MD \oplus SD_{\text{IV}(1)} = [-5 \;\; -5 \;\; -4 \;\; 0 \;\; 1 \;\; 2 \;\; 3 \;\; 4 \;\; 6 \;\; 16]$.

4. Performance Evaluation

4.1. Analysis of Connectivity. A random graph $G(n, p)$ is a graph of n nodes for which the probability that a link exists between two nodes is p. When p is zero, the graph does not have any edge, whereas when p is one, the graph is fully connected. Spencer [41] and Erdős and Rényi [42] showed that, for monotone properties, there exists a value of p such that the property moves from "nonexistent" to "certainly true" in a very large random graph. The function defining p is called the threshold function of the property. Given a desired probability P_c for graph connectivity, the threshold function p is defined by

$$\begin{aligned} P_c &= \lim_{n \to \infty} P_r\left[G(n, p) \text{ is connected}\right] = e^{e^{-c}}, \\ &\text{where } p = \frac{\ln(n) - \ln\left(-\ln\left(P_c\right)\right)}{n}. \end{aligned} \tag{23}$$

Let p be the probability that a shared key exists between two sensor nodes, and let n be the number of nodes, and d be the expected degree as

$$d = p \times (n-1) = \frac{(n-1)\left(\ln(n) - \ln\left(-\ln\left(P_c\right)\right)\right)}{n}. \tag{24}$$

For the deployment of a sensor network, let N be the expected number of neighbors within the communication range of a node. Using the expected node degree discussed above, the required local connectivity, P_{required}, can be estimated as follows:

$$P_{\text{required}} = \frac{d}{N} = \frac{(n-1)\left(\ln(n) - \ln\left(-\ln\left(P_c\right)\right)\right)}{nN}. \tag{25}$$

After the required local connectivity is obtained, the value S (the size of the key pool) and k (the number of keys in each node) are decided. Note that S is not directly related to the sensor network, while k is related to the memory size of a sensor node. Therefore, k needs to be as small as possible. Denote P_{actual}, the actual local connectivity, which is the probability of any two neighboring nodes to find a common key between themselves. The link availability of any two nodes of the existing schemes [15, 16] is obtained by $1-P_{\text{ns}}$, where P_{ns} is the probability that a pair of nodes do not share a common key. It can be found using P_{actual},

$$P_{\text{actual}} = 1 - \frac{{}_{S}C_{k} \times {}_{S-k}C_{k}}{\left({}_{S}C_{k}\right)^{2}} = 1 - \frac{((S-k)!)^{2}}{S!\,(S-2k)!}. \tag{26}$$

P_{actual} is then approximated as follows:

$$P_{\text{actual}} = 1 - \frac{(S-k)^{2S-2k+1}}{(S-2k)^{S-2k+(1/2)}}. \tag{27}$$

Recall that the Eschenauer and Gligor (E-G) [15] and Chan et al. (C-P) [16] schemes are the two representative key predistributions employing the random graph approach like the proposed scheme. Therefore, the efficiency of the proposed scheme is compared with these two schemes. Figure 3 compares the actual local connectivity of the proposed scheme with them when the size of the key varies from 2 to 200 for the S value of 5000 and 10000. Observe from the figure that the local connectivity increases as the number of keys in a node increases for the existing scheme when the size of the pool of keys is fixed. Note that the proposed scheme always allows the connectivity regardless of the number of keys per node. Also, the superiority of the proposed scheme becomes more substantial when the memory size of a sensor node is small.

4.2. Security of Connection. When the sizes of the key pool and key ring are S and K, respectively, the probability of two key rings sharing at least one key is calculated by (27). The number of cases each of two nodes chooses k keys out of the entire key pool is estimated in (28)

$$(C(S,k))^{2} = \left(\frac{S!}{k!\,(S-k)!}\right)^{2}. \tag{28}$$

Equation (29) estimates the number of cases where i shared keys are chosen from the entire key pool

$$C(S,i) = \frac{S!}{(S-i)!i!}. \tag{29}$$

The probability of two nodes sharing exactly i keys is calculated in (30)

$$P(i) = \frac{C(S,i)\,C(S-i,2(k-i))\,C(2(k-i),k-i)}{(C(S,k))^{2}}. \tag{30}$$

If a node of a wireless sensor network is captured by an adversary, the key information kept in the node might be exposed. The degree that the sensor nodes can operate while maintaining a desired security level is defined as the resilience against node capture. The higher the resilience against node capture, the longer the security level can be maintained during the operation of wireless sensor network.

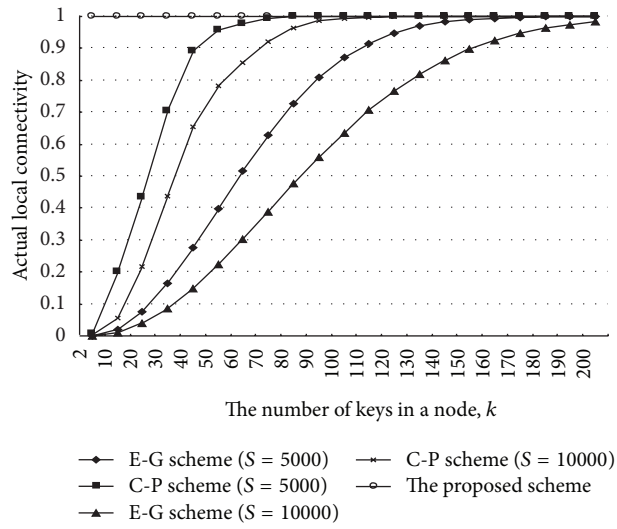

Figure 3: Comparison of connectivity of different schemes.

In the proposed scheme, the number of keys used for session key distribution is α, which is much larger than i, leading to a lower probability of a communication link being exposed, compared to the existing schemes. Equation (31) shows that the probability of the session key to be exposed leads to a large value when x number of nodes are captured with the proposed scheme

$$\left(1 - \left(1 - \frac{k}{S}\right)^{x}\right)^{\alpha}, \quad i < \alpha \le k. \tag{31}$$

The number of neighbor nodes sharing a key with the existing random graph based schemes is obtained by simulation to evaluate the security of the connection depending on the sizes of key rings. In the simulation the size of key rings is reduced in units of 100 each time, starting from 1,000 to 2,000, where the size of the key pool is 100,000. Then, the probability of establishing a secure connection between any source and destination node randomly selected is found by simulation. In addition, the hop count of the path required for secure connection is found to evaluate the efficiency of the scheme.

Figure 4 shows the average number of neighbor nodes as the size of key ring a node has changed, when the size of the key pool is 100,000 and 1,000 nodes deployed with a 50 m transmission zone in a $1,000 \times 1,000$ region. The number of neighbor nodes with the existing schemes represents the number of nodes sharing a key, whereas it does the ones which can communicate with each other with the proposed approach. Note that the proposed approach is not affected by the size of the key ring. Observe from the figure that a key is shared when the size of key ring exceeds 600 (E-G scheme) and 500 (C-P scheme), respectively. It is predicted that the security of the connection degrades as the size of the key ring

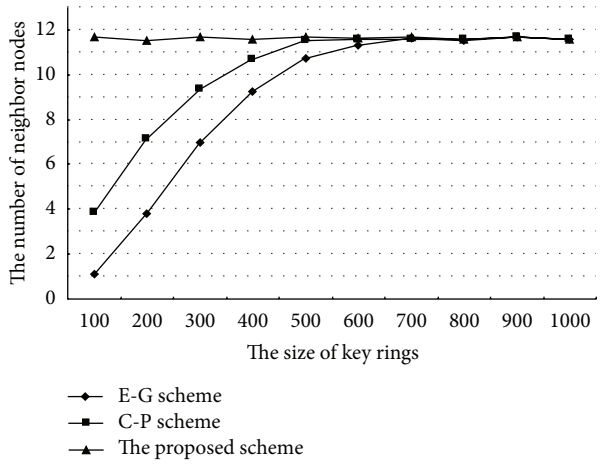

Figure 4: Comparison of the number of neighbor nodes sharing a key.

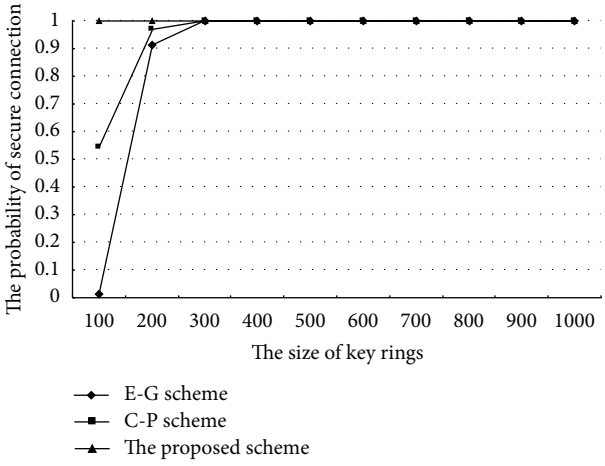

Figure 5: Comparison of the probability of secure connectivity depending on size of key ring.

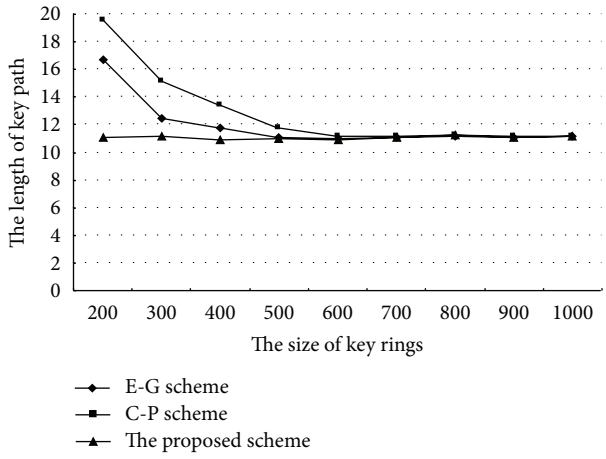

Figure 6: Comparison of key path length depending on the size of key ring.

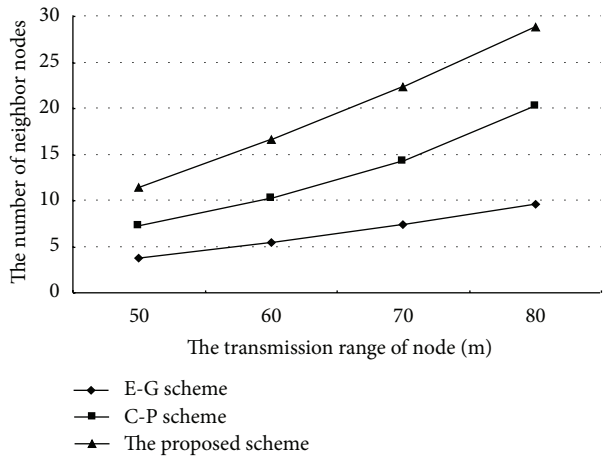

Figure 7: The number of neighbor nodes versus transmission range.

becomes smaller, causing the number of neighbor nodes to decrease dramatically.

Figure 5 shows the result of the evaluation of secure connectivity as the size of the key ring varies. It tests if secure connection is allowed between two randomly selected nodes. The proposed scheme always guarantees secure connectivity regardless of the size of the key ring, while the previous schemes do it only when the size exceeds around 300. The probability of secure connectivity drops sharply when the size falls below 300, for which the number of neighbor nodes is small.

Figure 6 shows the average length of the key path as the size of key ring varies. The length of the key path measures the distance from a node to the destination node where secure connection is allowed. When the size of a key ring is 100, the probability of secure connectivity is close to 0, and thus it is excluded from the test. The proposed scheme demonstrates consistent size of key path irrespective of the size of the key ring, whereas the existing schemes show similar performance to that of the proposed scheme when the size of the key ring exceeds 500. However, with the existing scheme, the lengths increase if the size of the key ring drops below 400.

Figure 7 shows the number of neighbor nodes as the transmission range of a node changes. As the transmission range increases, the number of neighbor nodes important for secure connectivity rises. This indicates that probability of secure connectivity grows in proportion to the transmission range. Figure 8 shows the length of the key path as the transmission range of a node varies. Observe from the figure that the difference between the proposed scheme and the previous ones decreases as the transmission range increases. That is, the increased transmission range reduces the overhead of providing secure connectivity and the required key. However, increasing the transmission range significantly increases energy consumption.

The neighbor nodes of a node can include the ones of multiple hop away to improve the probability of secure connectivity. However, it may increase the energy overhead in

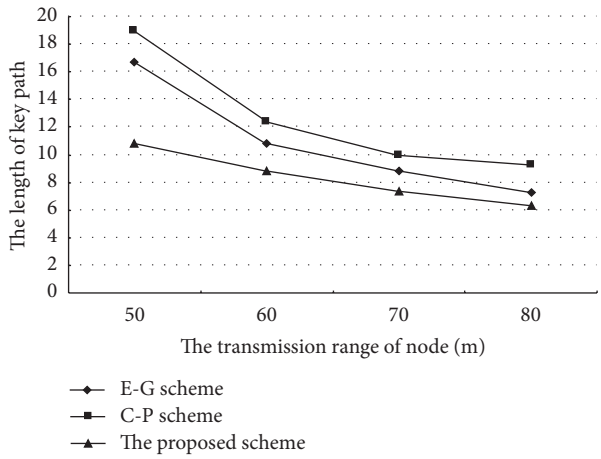

Figure 8: Comparison of key path length versus transmission range.

setting the keys and the number of nodes involved in setting the path key, resulting in lower security.

4.3. Energy Efficiency. We evaluate the energy efficiency of the proposed scheme through computer simulation by applying it to the LEACH protocol [43], which is one of the representative routing protocols proposed for WSNs. The number of cluster heads in the simulation is about 5% of the total number of nodes. We consider a sensor network of 100 sensor nodes randomly arranged in a 100×100 region. A base station is located at (50, 50).

In the simulation E_{elec} of the transmitter or receiver circuitry and ε_{amp} of the transmitter amplifier are set to 50 nJ/bit and 100 pJ/bit/m^2 [44–46], respectively, and initial residual energy of a sensor node is set to 25 J. The size of data packet is 3000 bits, the number of key rings of [15, 16] is 100, and key length is 32 bits. The energy consumption model [44, 46] is described as follows. For transmission, when a node transmits k-bit data to another node with distance d between them, the energy it consumes is

$$E_{\text{Tx}}(k, d) = E_{\text{elec}} \times k + \varepsilon_{\text{amp}} \times k \times d^2. \quad (32)$$

For receiving, when a node receives k-bit data, the energy it consumes is

$$E_{\text{Rx}}(k) = E_{\text{elec}} \times k. \quad (33)$$

The parameters used in the simulation are summarized in Table 1.

In the existing schemes each sensor node finds the neighboring nodes having the shared key for which energy consumption is not large. However, the energy consumption increases with the broadcasting of the identifier of the destination node via neighbor nodes holding a shared key for searching the pass key. If a node creates a pass key once and the session key is used more than once, the energy consumption may not greatly increase.

Table 1: The parameters used in the simulation.

Parameters	Value
Network size	100×100
Location of base station	(50, 50)
Number of nodes	100
Initial energy of the node	25 J
E_{elec}	50 nJ/bit
ε_{amp}	100 pJ/bit/m^2
Data size	3000 bits
Number of key rings	100
Key length	32 bits

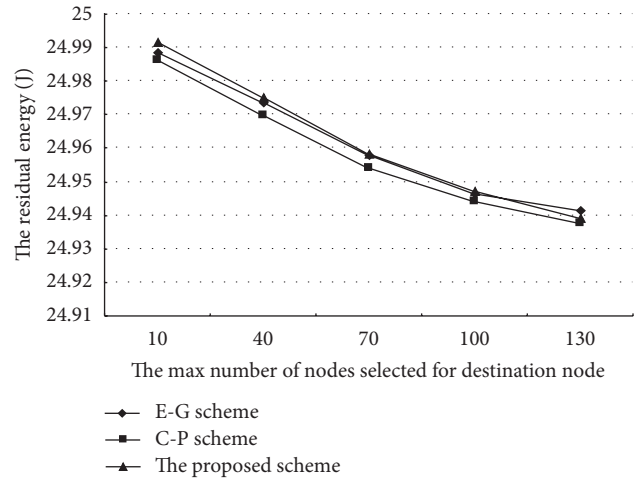

Figure 9: Comparison of the residual energy of the schemes.

Figure 9 compares the energy consumption when a node is randomly selected as the destination node out of some number of nodes (called destination group). Here only the energy used by the transceiver of a node taken for secure connection is considered. The energy consumed by the proposed scheme and previous schemes is low if the destination group is small. It gradually increases as the destination group increases. In the proposed scheme this is due to the increment of energy consumption taken to exchange the initial vector, while it owes to the energy consumed to set up the pass key in the previous scheme. The proposed scheme is affected by the size of destination group more significantly than the previous scheme, but it consumes less energy than the previous scheme in a typical environment where the size of destination group is usually small.

Figure 10 evaluates the energy consumption as time varies when the size of destination group is set to 100. The entire energy consumption is affected by the energy used to distribute the session key and the efficiency of the distribution of the session key. The increment of the energy consumption owes to session key distribution via the pass key in case of the previous schemes. The session key distribution of the previous scheme consumes more energy than the proposed scheme because the path for the pass key is longer.

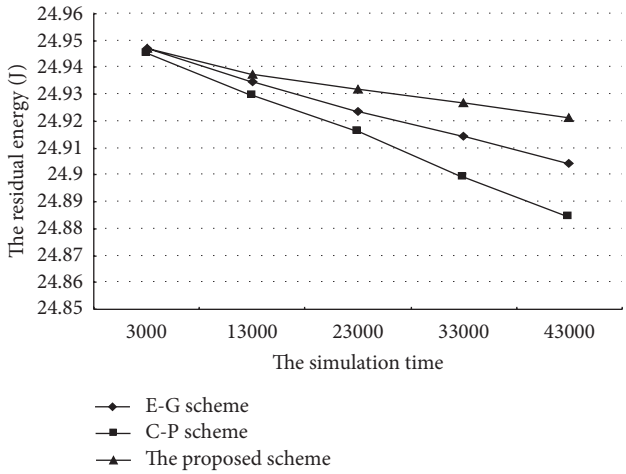

Figure 10: Comparison of the residual energy of the schemes as the key length varies.

5. Conclusion and Future Work

Most earlier schemes proposed for the security of wireless network used asymmetric cryptography such as the Diffie-Hellman key agreement or RSA. However, these schemes are inappropriate for wireless sensor networks due to the limited computation and energy resource of sensor nodes. In order to solve the problem the key distribution scheme using the trusted server was proposed based on asymmetric public key certification approach. Also, the key pre-distribution scheme that saves the key information before installing the sensor node was proposed, which is known to be very effective. The key pre-distribution scheme proposed by Eschenauer and Gligor creates various random keys in the base station, and the randomly selected keys are distributed to each sensor node. The sensor nodes having a common key between them use it as a mutual secret key. When they cannot create a path key, in other words when they cannot find a secret key, they cannot communicate with each other.

This paper has proposed a new key pre-distribution scheme guaranteeing that any pair of nodes can find a common secret key between themselves by using the keys assigned by eigenvalue and eigenvector of a square matrix of a pool of keys. Mathematical analysis and computer simulation revealed that the proposed scheme significantly reduces the overhead required for secure connectivity and energy consumption compared to the existing schemes. Analysis shows that the existing scheme requires a large number of keys in each sensor node to display a comparable connectivity as the proposed scheme. The probability of secure connectivity was evaluated by computer simulation for a randomly designated destination node, which reveals that the size of key ring of the existing schemes needs to be over 300 to be comparable with the proposed scheme. Also, the number of neighbor nodes having a shared key with the existing schemes needs to be over 600. The superiority of the proposed scheme is more substantial when the memory size of the sensor node is small.

When composing a network, the keys need to be pre-distributed as the nodes are deployed in the field. The effectiveness of the key pre-distribution scheme needs to be analyzed as new nodes are added. This is because the probability of secure connectivity may change as the network topology is varied. There is a need of application and performance analysis for the distributed sensor network model covering various conditions including the size of network. In order to raise the probability of secure connectivity, the transmission range of a node can be increased. However, the energy cost increases in proportion to the distance. A new approach considering the energy efficiency along with secure connectivity will also be investigated in the future. The proposed scheme capitalizes some important properties of matrix including eigenvalues and eigenvectors in generating a pool of random keys. It will be expanded to exploit some other properties which allow higher security at lower implementation and operation cost.

Acknowledgments

This research was supported in part by Korea Association of Industry, Academy and Research Institute (C0017380), DAPA and ADD (UD10070MD), Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1 A2040257), and the MSIP (Ministry of Science, ICT and Future Planning), Korea, in the ICT R&D Program 2013.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] D. Du, H. Xiong, and H. Wang, "An efficient key management scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 406254, 14 pages, 2012.

[3] S. Ruj and B. Roy, "Revisiting key predistribution using transversal designs for a grid-based deployment scheme," *International Journal of Distributed Sensor Networks*, vol. 5, no. 6, pp. 660–674, 2009.

[4] R. Zhu, Y. Qin, and J. Wang, "Energy-aware distributed intelligent data gathering algorithm in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 235724, 13 pages, 2011.

[5] R. kishore, S. Radha, and S. G. Hymlin Rose, "Wireless sensor network survey," *International Journal of Distributed Sensor Networks*, vol. 5, no. 6, pp. 850–866, 2009.

[6] A. Goyal, N. Kaur, Padmavati, Kuldeep, and R. Garimella, "Distributed energy efficient key distribution for dense wireless sensor networks," in *Proceedings of the 1st International Conference on Computational Intelligence, Communication Systems and Networks (CICSYN '09)*, pp. 143–148, July 2009.

[7] D. Liu, P. Ning, A. Liu, C. Wang, and W. K. Du, "Attack-resistant location estimation in wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 11, no. 4, pp. 1–22, 2008.

[8] A. Boukercha, L. Xua, and K. EL-Khatibb, "Trust-based security for wireless ad hoc and sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2413–2427, 2007.

[9] B. C. Neuman and T. Ts'o, "Kerberos. An authentication service for computer networks," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 33–38, 1994.

[10] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: security protocols for sensor networks," in *Proceedings of the 7th ACM/IEEE Annual International Conference on Mobile Computing and Networking*, pp. 189–199, July 2001.

[11] S. J. Choi, H. Y. Youn, and B. K. Lee, "An efficient dispersal and encryption scheme for secure distributed information storage," in *Proceedings of the International Conference on Computational Science*, pp. 958–967, 2003.

[12] R. Anderson and M. Kuhn, "Tamper resistance: a cautionary note," in *Proceedings of the 2nd Usenix Workshop on Electronic Commerce*, vol. 2, pp. 1–11, 2008.

[13] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pp. 52–61, 2003.

[14] C. Blundo, A. D. Santix, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pp. 471–486, 1993.

[15] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41–47, November 2002.

[16] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 197–213, May 2003.

[17] S. A. Camtepe and B. Yener, "Combinatorial design of key distribution mechanisms for wireless sensor network," in *Proceedings of the 9th European Symposium on Research in Computer Security (ESORICS '04)*, pp. 293–308, 2004.

[18] J. Lee and D. R. Stinson, "On the construction of practical key predistribution schemes for distributed sensor networks using combinatorial designs," *ACM Transactions on Information and System Security*, vol. 11, no. 2, pp. 1–35, 2008.

[19] D. Sánchez and H. Baldus, "A deterministic pairwise key predistribution scheme for mobile sensor networks," in *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm '05)*, pp. 277–288, September 2005.

[20] D. Chakrabarti, S. Maitra, and B. Roy, "A key pre-distribution scheme for wireless sensor networks: merging blocks in combinatorial design," *International Journal of Information Security*, vol. 5, no. 2, pp. 105–114, 2006.

[21] Z. Liu, J. Ma, Q. Huang, and S. Moon, "Asymmetric key predistribution scheme for sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1366–1372, 2009.

[22] H. T. T. Nguyen, M. Guizani, M. Jo, and E. N. Huh, "An efficient signal-range-based probabilistic key predistribution scheme in a wireless sensor network," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 5, pp. 2482–2497, 2009.

[23] P. Szczechowiak and M. Collier, "Practical identity-based key agreement for secure communication in sensor networks," in *Proceedings of the 18th International Conference on Computer Communications and Networks (ICCCN '09)*, pp. 1–6, August 2009.

[24] S. Banihashemian and A. G. Bafghi, "A new key management scheme in heterogeneous wireless sensor networks," in *Proceedings of the 12th International Conference on Advanced Communication Technology (ICACT '10)*, pp. 141–146, February 2010.

[25] F. Anjum, "Location dependent key management in sensor networks without using deployment knowledge," *Wireless Networks*, vol. 16, no. 6, pp. 1587–1600, 2010.

[26] A. Diop, Y. Qi, Q. Wang, and S. Hussain, "An efficient and secure key management scheme for hierarchical wireless sensor networks," *International Journal of Computer and Communication Engineering*, vol. 1, no. 4, pp. 365–370, 2012.

[27] X. He, M. Niedermeier, and H. Meer, "Dynamic key management in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 611–622, 2013.

[28] M. A. Simplicio Jr., P. S. L. M. Barreto, C. B. Margi, and T. C. M. B. Carvalho, "A survey on key management mechanisms for distributed wireless sensor networks," *Computer Networks*, vol. 54, no. 15, pp. 2591–2612, 2010.

[29] W. Bechkit, Y. Challal, A. Bouabdallah, and V. Tarokh, "A highly scalable key pre-distribution scheme for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 948–959, 2013.

[30] W. Gu, X. Bai, and S. Chellappan, "Scaling laws of key predistribution protocols in wireless sensor networks," Tech. Rep., The Department of Computer Science, Missouri University of Science and Technology, 2010.

[31] J. Kim, J. Lee, and K. Rim, "Energy efficient key management protocol in wireless sensor networks," *International Journal of Security and Its Applications*, vol. 4, no. 2, pp. 1–12, 2010.

[32] S. Guo and Z. Qian, "A compromise resilient pair wise rekeying protocol in large scale wireless sensor networks," in *Smart Wireless Sensor Networks*, vol. 18, pp. 316–326, InTechOpen, 2010.

[33] C. Alcaraz, J. Lopez, R. Roman, and H. Chen, "Selecting key management schemes for WSN applications," *Computers & Security*, vol. 31, no. 8, pp. 956–966, 2012.

[34] W. Yu and S. Wang, "Key pre-distribution using combinatorial designs for wireless sensor networks," *WSEAS Transactions on Mathematics*, vol. 12, no. 1, pp. 32–41, 2013.

[35] M. B. Paterson and D. R. Stinson, "A unified approach to combinatorial key predistribution schemes for sensor networks," *Designs Codes and Cryptography*, pp. 1–27, 2012.

[36] M. I. Salam, P. Kumar, and H. Lee, "An efficient key predistribution scheme for wireless sensor network using public key cryptography," in *Proceedings of the 6th International Conference on Networked Computing and Advanced Information Management (NCM '10)*, pp. 402–407, August 2010.

[37] T. D. Subash and C. Divya, "Novel key pre-distribution scheme in wireless sensor network," in *Proceedings of the International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT '11)*, pp. 959–963, March 2011.

[38] S. Mitra, R. Dutta, and S. Mukhopadhyay, "A hierarchical deterministic key pre-distribution for WSN using projective planes," in *Ad Hoc Networks*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 16–31, 2012.

[39] G. Nakos and D. Joyner, *Linear Algebra with Applications*, Brooks/Cole, Pacific Grove, Calif, USA, 1998.

[40] *Linear Algebra*, Birkhäuser, Boston, Mass, USA, 1997.

[41] J. Spencer, *The Strange Logic of Random Graphs*, Algorithms and Combinatorics, Vol. 22, Springer, Heidelberg, Germany, 2000.

[42] P. Erdős and A. Rényi, "On random graphs I," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.

[43] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii*

International Conference on System Siences (HICSS '00), pp. 323–327, Maui, Hawaii, USA, January 2000.

[44] K. T. Kim, B. J. Lee, J. H. Choi, B. Y. Jung, and H. Y. Youn, "An energy efficient routing protocol in wireless sensor networks," in *Proceedings of the 12th IEEE International Conference on Computational Science and Engineering (CSE '09)*, pp. 132–139, August 2009.

[45] N. Xiong, M. Cao, A. V. Vasilakos, L. T. Yang, and F. Yang, "An energy-efficient scheme in next-generation sensor networks," *International Journal of Communication Systems*, vol. 23, no. 9-10, pp. 1189–1200, 2010.

[46] K. T. Kim, C. H. Lyu, S. S. Moon, and H. Y. Youn, "Tree-based clustering(TBC) for energy efficient wireless sensor networks," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA '10)*, pp. 680–685, April 2010.