CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. (2015)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.3434

SPECIAL ISSUE PAPER

Self-healing wireless sensor networks

Atsuko Miyaji and Kazumasa Omote*,†

School of Information Science, JAIST, Nomi, Japan

SUMMARY

Availability is very important for long-term use of wireless sensor networks (WSNs), assuming the presence of an attacker. It is thus important to achieve secure communication among WSNs even if some sensor nodes are compromised. Self-healing WSNs possess the feature that a network automatically *self-heals* after node-capture attacks in order to achieve availability. The self-healing means that the ratio of compromised links decreases with time, even if the attacker corrupts sensor nodes of the network. In this paper, three kinds of self-healing schemes for WSNs are described, a polynomial-based self-healing scheme, a simple random key pre-distribution scheme with self-healing, and a proactive co-operative link self-healing scheme. Our contributions are the self-healing schemes with security evaluation, in which we conduct analytical evaluation and a simulation experiment of our schemes, and results obtained from both analysis and simulations indicate that our schemes are effective in self-healing. Furthermore, comparing three schemes, we clarify each difference and discuss optimal scheme under each different environments. © 2015 The Authors. *Concurrency and Computation: Practice and Experience* Published by John Wiley & Sons Ltd.

Received 29 November 2013; Revised 16 October 2014; Accepted 17 October 2014

KEY WORDS: wireless sensor networks; security; self-healing; resiliency

1. INTRODUCTION

1.1. Background

Wireless sensor networks (WSNs) are commonly used for military, smart homes, intelligent environments, and ubiquitous applications. The primary aim of WSNs is to sense some events and carry these sensor data to a base station. When the WSNs are deployed in hostile areas, sensor nodes can be captured by adversaries, and then information about the network is taken from the captured nodes, because a node has no tamper-resistant hardware. It is thus important to decrease the links compromised by node capture attacks. We describe an RKP (random key pre-distribution) scheme and *self-healing WSNs* in the following paragraphs.

Wireless sensor networks consist of small, battery-operated, limited memory, and limited computational power sensor nodes. Hence, most existing schemes in WSNs are based on symmetric key cryptography. One of the most popular schemes, referred to as *RKP* in this paper, was firstly proposed by Eschenauer and Gligor [1]. In this scheme, each node is configured with a key ring of m sub-keys. These keys are randomly drawn from the large key pool of P sub-keys. Two nodes establish their symmetric key from the sub-keys they have in common in their key ring. However, the security of the whole network in RKP degrades over time in hostile areas. An attacker who corrupts several nodes can partially reconstruct, from key rings of the compromised nodes, the key pool of

*Correspondence to: Kazumasa Omote, School of Information Science, JAIST, Nomi Japan.

†E-mail: omote@jaist.ac.jp

the system. If the attacker is continuously corrupting nodes, it will eventually learn the whole key pool, and all newly deployed nodes will establish links that will immediately be compromised. This is a non-desirable quality.

The WSNs are usually deployed to operate for a long period of time. Availability is very important for long-term use of WSNs under the presence of an attacker. Actually, we can find several schemes [2–7], which maintain availability of the secure link. We call these schemes self-healing WSNs, which possess the feature that a network automatically *self-heals* after node-capture attacks in order to achieve availability. The self-healing means that the ratio of compromised links decreases with time, even if the attacker corrupts sensor nodes of the network.

1.2. Related work

The RKP scheme was proposed by Eschenauer and Gligor [1]. In this basic probabilistic scheme, each sensor node randomly picks a set of sub-keys from a key pool before deployment, so that any two of the sensor nodes have a certain probability of sharing at least one common key. Chan, Perrig, and Song [8] further extended this idea and presented a μ-composite key pre-distribution scheme, in which any two sensors share at least μ pre-distributed sub-keys. Inspired by the basic RKP scheme and the polynomial-based key pre-distribution scheme [9], Liu, Ning, and Li [10] proposed a polynomial-based RKP scheme: it is a random subset assignment scheme, in which a polynomial pool is used, instead of using a key pool as in the previous approaches. The random subset assignment scheme assigns to each sensor node the secrets generated from a random subset of polynomials in the polynomial pool.

Castelluccia and Spognardi [2] have proposed the RKP scheme with self-healing property, named a robust key pre-distribution *(RoK)* scheme, for *multiphase WSNs*, in which a link self-heals against node-capture attacks by redeploying a sensor node (with the server's help) when the battery of a sensor is depleted. The RoK scheme improves the resiliency of the RKP scheme by limiting the lifetime of the keys, and by refreshing keys. Some recent schemes improve the resiliency of the RoK scheme. Yilmaz *et al.* [3] proposed a more resilient scheme than the RoK scheme to speed up the self-healing process. Kalkan *et al.* [4] proposed a zone-based RKP (Zo-RoK) scheme, which combines the best parts of Du *et al.*'s scheme [11] and the RoK and improves the resiliency of the RoK scheme. Ergun *et al.* [5] also proposed a more resilient scheme than the RoK scheme, called Random Generation Material key pre-distribution scheme. However, this scheme has a drawback about memory size that the size of a key ring basically depends on $m \times Gw$ (refer to notations in Section 2.1), while the size of key ring is $2m$ in the RoK scheme. A comparatively large value would be set to Gw, which denotes the maximum life of a sensor. Tian *et al.* [6] proposed a time-based key management scheme for multiphase WSNs, based on the RoK. However, this scheme does not evaluate the ratio of compromised links; it may not improve the resiliency of the RoK scheme. Recently, Das [12] proposed a random key establishment scheme for multiphase deployment, which is more resilient than the other existing random key distribution schemes. But it does not have the self-healing property.

On the other hand, there are several schemes with self-healing property without the help of server. As for self-healing of the secret key for the purpose of data survival, the proactive co-operative self-healing (POSH) scheme [13] and the distributed self-healing (DISH) scheme [14] have been proposed by Pietro *et al.* and Ma *et al.*, respectively. These schemes use key evolution and sensor cooperation to self-heal the secret key, which encrypts the sensed data on a sensor node, for the purpose of data survival.

1.3. Contribution

We summarize the three kinds of self-healing schemes for WSNs described in Figure 1—polynomial-based self-healing scheme (RPoK), simple random key pre-distribution scheme with self-healing (S-RKP), and proactive co-operative link self-healing (POLISH)—and clarify each difference from the point of view of environment for the usage:

1. **RPoK**[7] is a strongly resilient polynomial-based RKP scheme for WSNs. A private sub-key is not directly stored in each sensor node by applying the polynomial-based [10] scheme to the

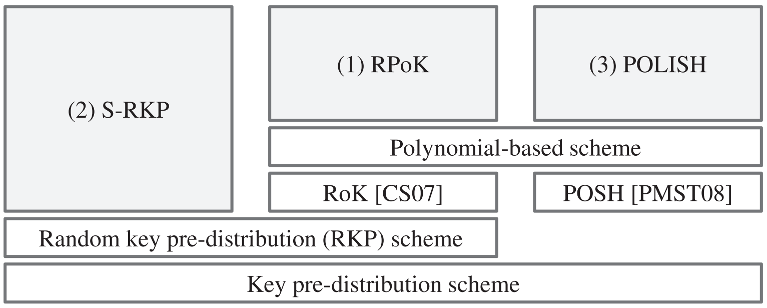

Figure 1. Map of our schemes. S-RKP, simple random key pre-distribution scheme with self-healing; RPoK, polynomial-based self-healing scheme; POLISH, proactive co-operative link self-healing.

RoK scheme [2]. As a result, RPoK is suitable in situations where higher resiliency is required such as a more hostile area.

2. **S-RKP**[15] is a simple RKP scheme with self-healing for WSNs, without lightweight operations such as a hash function. S-RKP can enhance the RKP with self-healing property, without changing the functions of sensors. This means that S-RKP is suitable for WSNs that use resource-poor sensors.
3. **POLISH**[16] is the first proactive co-operative link self-healing scheme, without the help of a server. POLISH is suitable for the situations where 100% secure connectivity is required and the size of memory is quite efficient in POLISH, because it is a deterministic key-sharing scheme. POLISH can also keep higher resiliency without the help of a server, where the sensor operates independently. Hence, POLISH is suitable for WSNs where the key management is not necessary by a server.

1.4. Organization

The rest of this paper is organized as follows: In the next section, we provide some preliminaries. We explain our protocols in detail in Section 3, analyze its security in Sections 4 and 5, and discuss security and efficiency of our schemes in Section 6. We finally conclude this paper in Section 7.

2. PRELIMINARIES

2.1. Notation

We use the common notations in Table I.

2.2. Requirements

The following requirements need to be considered when designing a self-healing scheme in WSNs.

Highly secure connectivity: After deployment, two nodes share a key to establish a secure link. A probabilistic key-sharing scheme is required to keep the probability of key sharing high. This probability is called a secure connectivity. Actually, in the RKP schemes, a secure connectivity becomes almost 100% by adjusting P and m.

Self-healing: Sensor nodes may be deployed in public or hostile locations in many applications. We assume that the adversary can mount a physical attack on a sensor node after it has been deployed and read secret information from its memory. Therefore, a self-healing property is very important for long-term use of WSNs. Self-healing means that the compromised links are automatically healed with time even if the adversary corrupts the sensor nodes of the network. The degree of self-healing is measured by *resiliency*. Resiliency is estimated by the ratio of links, which has not been compromised by the capture of nodes. Self-healing is achieved by security

Table I. Notation.

Symbol	Explanation
n	Total number of sensors (i.e., size of network)
s_A, ID_A	Sensor s_A and index of s_A
λ	Total sub-key space
P	Number of sub-keys in the key pool, which is a set of sub-keys randomly chosen from λ ($P \ll \lambda$)
m	Length of key ring on a sensor ($m \ll P$)
r	Round index (i.e., fixed-length time slot)
R	Number of rounds in one generation
c	Number of nodes captured in one round
Gw	Generation window (i.e., maximum lifespan of a sensor)
δ	Renewal ratio of the key pool at every round
KP^r, KR^r_A	Key pool and key ring of s_A, which is deployed at round r
k^r_ℓ	ℓth sub-key $\in KP^r$
q	Large prime number
H	Cryptographic hash function, which is one-way and collision-resistant, $H : \{0,1\}^* \rightarrow \{0,1\}^q$
F	Hash function $F : \{0,1\}^* \rightarrow \{0,1\}^{log_2(P)}$
$f^j_s(x,y)$	sth bivariate t-degree polynomial at generation j over a finite field $\mathbb{F}_q$
FKP^j, BKP^j	Forward and backward key pool at generation j
PLP^j	Polynomial pool at generation j
FKR^j_X	Forward key ring of X at generation j
BKR^j_X	Backward key ring of X at generation j
PLR^j_X	Polynomial ring of X at generation j
fk^j_s, bk^j_s	sth forward key $\in FKP^j$ and backward key $\in BKP^j$ at generation j
w	Number of links with neighboring sensors
$K^r_{i,j}$	Pairwise symmetric key (secure link) between s_i and s_j at round r
S^r_i, $c^r_{i_\ell}$	Seed of s_i and ℓth *contribution* received by s_i at round r
G^r, Y^r, R^r	Set of green, yellow, and red sensors at round r
GL^r, RL^r	Set of green and red links at round r

properties: *forward* and *backward security*. These security properties are defined in [13]. Forward secrecy means that adversary cannot learn any keys used to decrypt and/or authenticate before compromise, and backward secrecy means that adversary cannot learn any keys used to decrypt and/or authenticate after compromise.

Restricted resources: It is required that WSNs consist of small, battery-operated devices with limited memory and limited computational power. It is also desirable that we do not use even lightweight operations on a sensor, such as a hash function.

2.3. Attacker model

The main purpose of attacker is to steal as many keys in each node as possible in order to compromise a secure link. We assume eager attackers described in [2]. This type of attacker regularly corrupts nodes of the network without stopping operations. More concretely, the eager attacker keeps compromising nodes at a constant rate, from the deployment of the first round of sensors to the end of the life of the network. Attacker knows the entire topology of the WSNs and can create a table of sensor secrets and share them. Attacker does not stay at one local place for stealthy operation and then does not interfere with sensor's behavior, that is, it does not delete, delay, or introduce messages.

2.4. Multiphase wireless sensor networks

Multiphase WSNs: A multiphase WSN is a network where a sensor is replaced with the server's help after its battery has been depleted. More concretely, sensor nodes that run out of power will be removed from the network, and new sensor nodes need to be periodically deployed to assure network connectivity. Note that a multiphase WSN does not always have self-healing property.

Resilient multiphase WSNs: A resilient multiphase WSN possesses the feature that the network automatically self-heals against node-capture attack. The key pool refreshes over time in resilient multiphase WSNs, and hence the pre-distributed keys have limited lifetimes. The key ring also refreshes over time. This implies that each sensor gradually stops using the old sub-keys.

2.5. *Probability of pairwise key sharing*

In the RKP schemes [1, 8], a pairwise key is stochastically constructed. The probability that two nodes share i sub-keys is defined as

$$p_i - \frac{\binom{P}{i}\binom{P-i}{2(m-i)}\binom{2(m-i)}{m-i}}{\binom{P}{m}^2}, \tag{1}$$

where m is the key ring size and P is the key pool size. Therefore, the probability that two nodes share at least one sub-key is defined by $1 - p_0$.

2.6. *Polynomial-based scheme*

We briefly review the basic polynomial-based key pre-distribution protocol [9]. Because our goal is to establish pairwise keys, for simplicity, we only discuss the special case of pairwise key establishment in the context of sensor networks. To pre-distribute pairwise keys, a setup server randomly generates t-degree $f(x, y)$ over a finite field $\mathbb{F}_q$, where it has the symmetrical property of $f(x, y) = f(y, x)$. The security proof in [9] ensures that this scheme is unconditionally secure and t-collusion resistant. That is, the collusion of no more than t compromised sensor nodes learns nothing about the pairwise key between any two non-compromised nodes.

It is assumed that each sensor has a unique ID. For a sensor A, the setup server computes a polynomial share of $f(x, ID_A)$. For any two sensor nodes A and B, node A can compute the common key $f(ID_B, ID_A)$ by evaluating $f(x, ID_A)$ at point ID_B, and node B can compute the same key $f(ID_A, ID_B) = f(ID_B, ID_A)$ by evaluating $f(x, ID_B)$ at point ID_A. The sensor node A needs to store a t-degree polynomial $f(x, ID_A)$.

As explained in [10], the average probability that a link is indirectly compromised at generation j is given by

$$P_{Poly}(j) = 1 - \sum_{i=0}^{t} \binom{cj}{i} \left(\frac{m}{P}\right)^i \left(1 - \frac{m}{P}\right)^{cj-i} \tag{2}$$

3. SELF-HEALING WIRELESS SENSOR NETWORK PROTOCOLS

System and network assumptions: Time is divided into equal and fixed rounds. In RPoK and POLISH schemes, round synchronization can be implemented, but, in S-RKP, round synchronization is not necessary to be implemented in a node. The network is connected at all times. Any two sensors can communicate either directly or indirectly, via other sensors. In RPoK and POLISH, each sensor can perform cryptographic hashing and polynomial execution, but, in S-RKP, no sensor performs cryptographic hashing or polynomial execution (same as the RKP schemes).

3.1. *Polynomial-based self-healing scheme*

The primary aim of RPoK is to not only increase secure connectivity between nodes but also decrease the compromised ratio of nodes against node-capture attacks in multiphase WSNs. Practically, a private sub-key is not directly stored in each sensor node by applying the t-degree polynomial-based scheme to the RoK scheme [2]. As a result, an attacker has to capture $(t + 1)$

sub-keys in order to corrupt a link. Furthermore, we achieve the forward and backward security of the polynomial by linear transformations using forward and backward keys. Therefore, RPoK can dramatically improve the ratio of compromised links compared with the RoK scheme.

Protocol description: The protocol details of RPoK are as follows:

1. **Pool generation**. RPoK uses three kinds of pools, that is, FKP^j, BKP^j, and PLP^j, where FKP^j and BKP^j are the same as RoK. PLP^j is defined as $PLP^j = \left\{f_1^j(x,y), f_2^j(x,y), \ldots, f_{P/2}^j(x,y)\right\}$, where $f_s^j(x,y) = \alpha_{j-1} f_s^{j-1}(x,y) + \beta_{j-1}$, $\alpha_{j-1} = H\left(fk_s^{j-1} \parallel bk_s^{j-1}\right)$ and $\beta_{j-1} = H\left(bk_s^{j-1} \parallel fk_s^{j-1}\right)$ ($j = 1, \ldots, N$, $s = 1, \ldots, P/2$).
2. **Ring assignment**. Node A is configured with key rings, defined as $FKR_A^j = \left\{fk_s^j\right\}$, $BKR_A^j = \left\{bk_s^j\right\}$, and $PLR_A^j = \left\{f_s^j(x,y)\right\}$, such that $s = F(ID_A \parallel i \parallel g_A)$ ($i = 1, 2, \ldots, m/2$). Note that $g_A = j$ when the node A is deployed at generation j.
3. **Establishing a secure link**. After deployment, a node A initiates neighbors discovery procedure with node B, and both nodes calculate indices, similar to RoK. If there are collisions such that $F(ID_B \parallel y \parallel g_B) = F(ID_A \parallel x \parallel g_A)$, where $x, y \in \{1, 2, \ldots m/2\}$, then it is known that they both have $fk^{g_B}_{F(ID_B \parallel y \parallel g_B)}$, $bk^{g_A+Gw-1}_{F(ID_B \parallel y \parallel g_B)}$, and $f^{g_B}_{F(ID_B \parallel y \parallel g_B)}(ID_A, ID_B)$ in their memory. In this way, all colluding local indices $a, b, \ldots, z \in \{1, 2, \ldots m/2\}$ are found, and the following becomes their pairwise symmetric key:

$$K_{AB}^{RPoK} = H\left(f^{g_B}_{F(ID_B \parallel a \parallel g_B)}(ID_A, ID_B) \parallel \cdots \parallel f^{g_B}_{F(ID_B \parallel z \parallel g_B)}(ID_A, ID_B)\right) \quad (3)$$

   Note that $f_s^{g_B}(ID_A, ID_B)$ satisfies both forward and backward security because of linear transformations, as mentioned in the pools generation phase. Furthermore, in RPoK, K_{AB}^{RPoK} is a session key in each round, while K_{AB}^{RoK} in RoK is a common key in the overlapping generations. We assume that K_{AB}^{RPoK} is updated in each round.

3.2. *Simple random key pre-distribution scheme with self-healing*

In order to attach self-healing property to the RKP scheme, all the previous RKP schemes with self-healing property had to change the process of each sensor as well as that of a server. On the other hand, the primary aim of S-RKP is to attach self-healing property to the RKP scheme by simply changing only the server process. The S-RKP can attach self-healing property to existing RKP schemes by updating the key pool of a server with time. The most interesting point of this scheme is that processing of each sensor is the same as in the RKP scheme, that is, round synchronization is not necessary to be implemented in a node. We emphasize that the keys, which can be assigned to a sensor are not updated, same as the RKP scheme. Nevertheless, the keys have a limited lifetime, similar to the RoK scheme. S-RKP takes a different approach from the RoK scheme. Thanks to such a server process in the self-healing RKP, a sensor does not use even lightweight operations such as a hash function.

3.2.1. Protocol description. This protocol is quite simple. Some additional executions by a server are required, while no additional execution on a node is necessary. The procedure on each sensor is the same as in the RKP scheme. The key pool in the RKP scheme is composed of random keys that do not evolve with time. In contrast, the key pool is composed of random keys that the server evolves with time in S-RKP.

1. **Pool generation**. A server sets the key pool in this protocol. In order to generate the key pool, the server uses a hash chain using a cryptographic hash function H and a seed s. The key pool is initiated with P random sub-keys. Let k_ℓ^r be the ℓth key at round r in the key pool. A server computes the sub-keys $k_1^0 = H(s||1)$, $k_2^0 = H\left(s||k_1^0\right)$, …, $k_P^0 = H\left(s||k_{P-1}^0\right)$ in the key pool. Thus, the key pool at round 0 (i.e., when the network is first deployed) is defined as

$KP^0 = \{k_1^0, k_2^0, \ldots, k_P^0\}$, where k_1^0 is the oldest sub-key at round 0. KP^0 corresponds to the key pool of the RKP scheme.

2. **Pool update**. At each round, the key pool is partially updated over time. Let δ be the renewal size of the key pool at every round. For instance, if $\delta = 1$ at round 1, then $k_1^1 \leftarrow k_2^0, k_2^1 \leftarrow k_3^0$, …, $k_{P-1}^1 \leftarrow k_P^0$, $k_P^1 \leftarrow H\left(s||k_{P-1}^1\right)$. Generally, the keys are partially renewed at round r as follows:

$$KP^r = \{k_1^r, k_2^r, \ldots, k_{P-\delta}^r, k_{P-\delta+1}^r, \ldots, k_{P-1}^r, k_P^r\}, \tag{4}$$

where $k_1^r = k_{1+\delta}^{r-1}$, $k_2^r \leftarrow k_{2+\delta}^{r-1}$, …, $k_{P-\delta}^r \leftarrow H\left(s||k_{P-\delta-1}^r\right)$, $k_{P-\delta+1}^r \leftarrow H\left(s||k_{P-\delta}^r\right), \ldots,$ $k_{P-1}^r \leftarrow H\left(k_{P-2}^r\right)$, $k_P^r \leftarrow H\left(k_{P-1}^r\right)$. Because the key pool slides just δ at every round, all the keys in the key pool are replaced after $\lceil P/\delta \rceil$ rounds. A server manages the current P sub-keys in the key pool. The server discards the old sub-keys.

3. **Ring assignment**. This step is the same as ring assignment in the RKP scheme. For each sensor, m keys are randomly selected from the current key pool and stored in the sensor's memory before deployment. This set of m keys is called the sensor's key ring.
4. **Establishing a secure link**. This step is also the same as the establishment of a secure link in the RKP scheme. After the sensors are deployed, s_i initiates key-discovery procedure with their neighbors with whom they share a key. Sensors, which discover that they contain a shared key in their key rings can then verify that their neighbors actually hold the key through a challenge-response protocol. Of course, this protocol can use 'path keys', as used in the RKP scheme.

3.3. Example

This section illustrates our protocol with an example (Figure 2). Let $(P, m, \delta, s) = (5, 3, 1, 7)$ and also let $\lambda = \{1, 2, \ldots, 100\}$ be the key space. We assume that three sensors $\{s_1, s_2, s_3\}$ are deployed at the beginning of round 0 and that s_3 is replaced at the beginning of round 2. At the first round 0, the key pool is initiated with $KP^0 = \{41, 13, 18, 75, 34\}$, where $41 = H(7||1)$, $13 = H(7||41)$, $18 = H(7||13)$, $75 = H(7||18)$ and $34 = H(7||75)$. At the next round, the key 41 is removed, and then the new key 22 is added as k_5^1, that is, $KP^1 = \{13, 18, 75, 34, 22\}$, where $22 = H(7||34)$. In this way, the key pool is partially updated at every round.

The key ring of each sensor at round 0 is selected from KP^0. Let $KR_1^0 = \{41, 13, 75\}$ be the key ring of s_1. When s_3 is replaced at the beginning of round 2, KR_3^2 is selected from KP^2. If $KR_3^2 = \{75, 34, 19\}$, then s_1 and s_3 share one common key, 75, to establish a secure link at round 2.

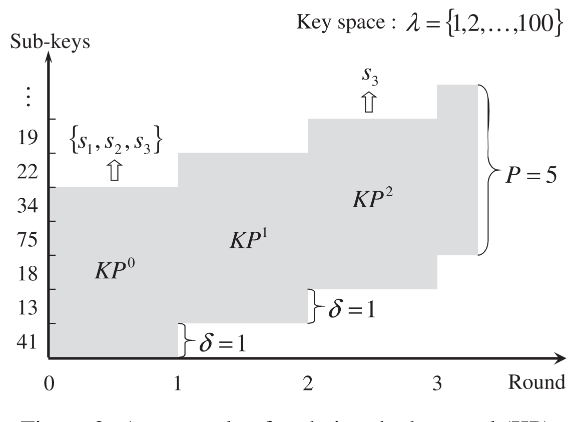

Figure 2. An example of updating the key pool (KP).

3.4. *Proactive co-operative link self-healing scheme*

To evaluate the healing rate of secret key for data encryption, the POSH scheme analyzes the number of green sensors in any round. The secret key K_i^r is used as a secure link between a sensor s_i and the sink at round r because the sink knows all the secret keys of sensors. However, we cannot directly achieve the secure link between sensors by the POSH scheme, because the security of a link between sensors is not considered in the POSH scheme.

In this section, we describe the POLISH scheme. The primary aim of this scheme is to decrease the compromised ratio of links against node-capture attacks without help of a server, that is, links compromised in WSNs automatically self-heal with time. POLISH updates a link using the random data transmitted from the neighboring sensors, based on the idea of the POSH scheme. Although this protocol is very simple like POSH, more importantly, our security evaluation is not achieved easily, that is, it is necessary to newly take the security of a link between sensors into consideration in POLISH because such security is not considered in the POSH scheme.

A link self-heals in two steps: first two neighboring sensors are self-healed, and then the link between these sensors is self-healed. A major difference between POSH and POLISH is the security analysis of a link. While the POSH scheme in a sense treats the secure link between a sensor and a powerful sink, POLISH treats the secure link between sensors. In addition, POLISH uses a bivariate t-degree polynomial, and thus an attacker has to capture $(t+1)$ polynomial shares during a limited period of time (i.e., at round 1) in order to corrupt a link.

An adversary breaks into $c = |R^r|$ sensors to read the pairwise symmetric keys and secret seeds of a Pseudo-Random Number Generator (PRNG) in R^r and to monitor all the communication of R^r. At any time, we identify three sets of sensors (i.e., green, yellow, and red) and two sets of links as follows:

- *Red links* (RL^r) are those that have been compromised in some round $r' < r$, and the pairwise symmetric key of the link is known to adversary in round r.
- *Green links* (GL^r) are those that have either never been compromised or regained their security in round r.

Note that, in this scheme, a red sensor s_i at round r means that adversary knows a seed S_i^r. If s_i becomes red in round r' and is self-healed at the end of round $r > r'$, then adversary can compute the contributions of s_i from round r' to r.

3.4.1. Protocol description. The protocol details of POLISH are as follows:

1. **Setup**. To pre-distribute pairwise keys, the setup server randomly generates a bivariate t-degree polynomial $f(x, y)$ over a finite field $\mathbb{F}_q$, such that it has the property of $f(x, y) = f(y, x)$. For each sensor s_i, the setup server computes a polynomial share of $f(x, y)$, that is, $f(x, ID_i)$. Each sensor can use a secure hash function, a polynomial, and a PRNG with a unique secret seed. Note that the secure degree t of polynomial is dependent on the number of adversary at each round. For instance, if we set $t \geqslant 10$ as the secure degree of polynomial when we assume $c = 10$ then adversary cannot recover $f(x, y)$.
2. **Establishing a secure link**. For any two sensors s_i and s_j, the sensor s_i can compute the key $f(ID_j, ID_i)$ by evaluating $f(x, ID_i)$, and the sensor s_j can compute the same key $f(ID_i, ID_j) = f(ID_j, ID_i)$ by evaluating $f(x, ID_j)$. As a result, sensors s_i and s_j can establish a pairwise symmetric key $K_{i,j}^1 = f(ID_i, ID_j)$ in the first round (round 1). After key establishment, s_i deletes all the coefficients of a polynomial.
3. **Key and seed update**. The neighboring sensors s_i and s_j have a pairwise symmetric key $K_{i,j}^1$ (secure link) when they are deployed at the beginning of the first round (round 1). At the beginning of round r, s_i produces w pseudo-random values (contributions) using its PRNG for w neighboring sensors and sends them to the neighboring sensors using a secure link. Note that all the contributions that s_i sends are different. Then, each sensor receives contributions from the neighboring sensors during round r. The recipient uses two contributions as inputs to the secure hash function used for key update. To update the secure link at the end of round r, s_i computes

$$K_{i,j}^{r+1} = H\left(K_{i,j}^{r}\left\|c_{i_\eta}^{r}\right\|c_{j_\lambda}^{r}\right), \tag{5}$$

where $c_{i_\eta}^{r}$ is the ηth contribution that s_i received at round r and $c_{j_\lambda}^{r}$ is the λth contribution that s_j received at round r. Both s_i and s_j delete $K_{i,j}^{r}$ after key updating. Furthermore, each sensor updates a seed of PRNG using w contributions, which are all contributions received by the neighboring sensors. To update the seed S_i^r at the end of round r, s_i computes[‡]

$$S_i^{r+1} = H\left(S_i^r\left\|c_{i_1}^r\right\|\cdots||c_{i_w}^r\right) \tag{6}$$

After seed updating, s_i deletes S_i^r. A seed is updated in every round, and then w contributions are generated by PRNG with such new seed.

Remark
In the POSH scheme, each sensor receives contributions from sensors, which are randomly chosen in WSNs. On the other hand, in POLISH, each sensor receives contributions from neighboring sensors. The probability that a contribution will be intercepted on the way by an adversary may become high in the POSH scheme, because a contribution can be sent from a sensor, which is far from the recipient.

3.4.2. The link state. A link self-heals in two steps: first two neighboring sensors are self-healed, and then the link between them is self-healed. We can generate the seven kinds of link states as described in Figure 3 (i.e., GL^r = {G(G)G} and RL^r = {G(R)G, G(R)Y, $Y(R)Y$, Y(R)R, G(R)R, R(R)R}). A pair of sensors and their common link constitutes a link state. For example, G(R)Y means that two neighboring sensors of green and yellow are connected by the red link. The conditions of transition are as follows:

1. *Double-compromised* condition means that both of two neighboring sensors are compromised.
2. *Single-compromised* condition means that either of two neighboring sensors is compromised.
3. *None-compromised* condition means that neither of two neighboring sensors is compromised.

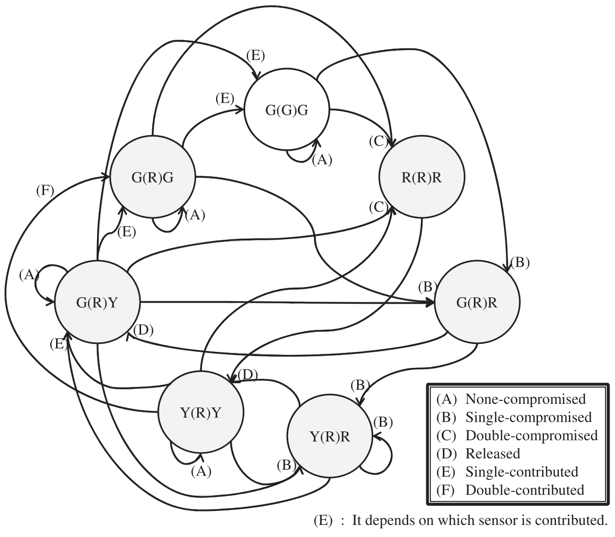

Figure 3. Link state transition diagram.

[‡]The update of a PRNG seed is similar to [17].

4. *Single-contributed* condition means that either of two neighboring sensors receives at least one 'secure contribution'.
5. *Double-contributed* condition means that both of two neighboring sensors receive at least one secure contribution.

Note that the secure contribution is a green contribution that is not intercepted by adversary.

A red link remains red if a red sensor is within the wireless communication range of both of two sensors, which constitute the red link. On the other hand, a green link remains green as long as both of two sensors, which constitute the green link are green. We notice that even if two sensors are green, the link between them can be also red (i.e., G(R)G). A green link (G(G)G) can be changed from two states G(R)G and G(R)Y when single-contributed. G(R)G becomes G(G)G when one of two neighboring sensors receives a secure contribution from the other. G(R)Y becomes G(G)G when the yellow sensor Y receives a secure contribution from this green sensor G. G(R)Y becomes G(G)G when Y receives at least one secure contribution from other green sensors except this G.

4. SECURITY EVALUATION BY ANALYTICAL MODEL

4.1. Secure connectivity

It is important to raise the secure connectivity, under strengthening resiliency. The higher the secure connectivity is, the better the self-healing scheme is. A self-healing scheme can be divided into two schemes, a deterministic and probabilistic key-sharing schemes. In this paper, POLISH is a deterministic key-sharing scheme, but RPoK and S-RKP are probabilistic key-sharing scheme. In a probabilistic key-sharing scheme, there is a tradeoff between secure connectivity and resiliency against node-capture attacks. RPoK and S-RKP establish 'almost certain' shared-key connectivity.

RPoK: The secure connectivity of RPoK is the same as that of the original RKP [1], that is, the probability that two nodes share i sub-keys is the same as Equation (1) as follows:

$$p_{i,RPoK} = \frac{\binom{P}{i}\binom{P-i}{2(m-i)}\binom{2(m-i)}{m-i}}{\binom{P}{m}^2} \tag{7}$$

The probability that two nodes share at least one sub-key is defined by $1 - p_{0,RPoK}$.

S-RKP: The secure connectivity of S-RKP is a little inferior to that of the original RKP, RoK, and RPoK because of the pool update. However, S-RKP can achieve secure key sharing without using the security executions. The δ sub-keys in the key pool are updated at every round as described in Section 3.2. Let KP^{r_1+r} be the key pool after r rounds from the key pool KP^{r_1} (r is a positive number). Hence, $KP^{r_1} - \left(KP^{r_1} \cap KP^{r_1+r}\right) = \delta r$ holds. We assume that a node A is deployed at round r_1, and then a node B is deployed after r rounds (refer to Figure 4). r means the difference in rounds between two deployments of nodes A and B. Note that the key rings of nodes A and B are randomly selected from KP^{r_1} and KP^{r_1+r}, respectively. The $p_{(i,r)}$ is defined as the average probability that the nodes A and B share i sub-keys when the difference in rounds is r. If nodes A and B are deployed at the same round (i.e., $r = 0$), the probability that two nodes share i sub-keys is the same as in Equation (1), defined by

$$p_{(i,0)} = \frac{\binom{P}{i}\binom{P-i}{2(m-i)}\binom{2(m-i)}{m-i}}{\binom{P}{m}^2} \tag{8}$$

If $r \geqslant 1$, then $p_{(i,r)}$ can be considered using Figure 4 as follows: The number of combination, which assigns sub-keys to nodes A and B is $\binom{P}{m}^2$ because $|KP^{r_1}| = \left|KP^{r_1+r}\right| = P$. Also, the number of combination of common i sub-keys from a set of $(P - \delta r)$ is $\binom{P-\delta r}{i}$. Here, we focus on assignment of the key ring of node A. Let x be the number of sub-keys of node A assigned from a set of $(P - \delta r - i)$ in KP^{r_1}, that is, x is the number of sub-keys of

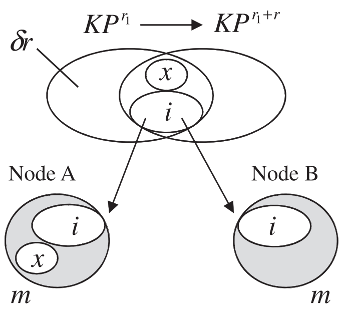

Figure 4. Conceptual diagram of analytical model. KP, key pool.

node A, which does not overlap with sub-keys of node B in $KP^{r_1} \cap KP^{r_1+r}$. Hence, the number of combination, which assigns sub-keys to the node A (excluding i sub-keys) from a set of $(P - \delta r - i)$ in KP^{r_1} is $\binom{P-\delta r-i}{x}$. Also, the number of combination, which assigns sub-keys to node A (excluding i and x sub-keys) from δr is $\binom{\delta r}{m-i-x}$. Furthermore, the number of combination, which assigns the rest of sub-keys of KP^{r_1+r} to node B (excluding i sub-keys) is $\binom{P-i-x}{m-i}$. Based on the previous details, the following equation is obtained by taking x into consideration from 0 to $m - i$:

$$p_{(i,r)} = \frac{\binom{P-\delta r}{i} \cdot \sum_{x=0}^{m-i} \binom{P-\delta r-i}{x} \binom{\delta r}{m-i-x} \binom{P-i-x}{m-i}}{\binom{P}{m}^2} \quad (r \geqslant 1), \tag{9}$$

where $\binom{a}{b} = 0$ when $a < b$ or $b < 0$.

$p_{(i,r)}$ (Equation (9)) should be an extension of $p_{(i,0)}$ (Equation (8)). The following theorem shows that $p_{(i,r)}$ is an extension of $p_{(i,0)}$:

Theorem 1

$p_{(i,r)}$ is defined for all $r \geqslant 0$ as follows:

$$p_{(i,r)} = \frac{\binom{P-\delta r}{i} \cdot \sum_{x=0}^{m-i} \binom{P-\delta r-i}{x} \binom{\delta r}{m-i-x} \binom{P-i-x}{m-i}}{\binom{P}{m}^2} \tag{10}$$

Proof

$p_{(i,r)}$ is defined when $r \geqslant 1$ in Equation (9). So, we show that $p_{(i,r)}$ satisfies Equation (8).

$$\begin{aligned} p_{(i,r=0)} &= \frac{\binom{P}{i} \sum_{x=0}^{m-i} \binom{P-i}{x} \binom{0}{m-i-x} \binom{P-i-x}{m-i}}{\binom{P}{m}^2} \\ &= \frac{\binom{P}{i}\binom{P-i}{m-i} \cdot 1 \cdot \binom{P-m}{m-i}}{\binom{P}{m}^2} \\ &= \frac{\binom{P}{i} \cdot \frac{(P-i)!}{(P-m)!(m-i)!} \cdot \frac{(P-m)!}{(P-2m+i)!(m-i)!}}{\binom{P}{m}^2} \\ &= \frac{\binom{P}{i} \cdot \frac{(P-i)!}{(P-2m+i)!(2m-2i)!} \cdot \frac{(2m-2i)!}{(m-i)!(m-i)!}}{\binom{P}{m}^2} \\ &= \frac{\binom{P}{i}\binom{P-i}{2m-2i} \cdot \binom{2m-2i}{m-i}}{\binom{P}{m}^2} = p_{(i,0)} \end{aligned}$$

□

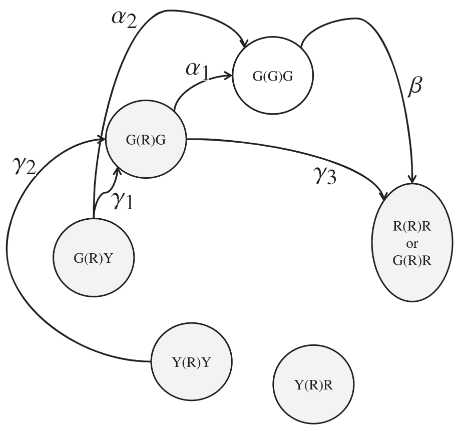

Figure 5. Partial link state transition diagram.

By estimating an appropriate r, we can derive $p_{i,S-RKP}$, which is the expected value of key-sharing probability based on $p_{(i,r)}$. We assume that node A is newly deployed at round r_2 and that the neighboring nodes of node A were deployed at round r_1 on an average $(r_1 < r_2)$. $p_{i,S-RKP}$ of node A is related to the difference in rounds between node A and its neighboring nodes. Therefore, $p_{i,S-RKP}$ is defined by

$$p_{i,S-RKP} = p_{(i,(r_2-r_1))} \tag{11}$$

When an arbitrary node in WSNs is taken up, the average age (generation) of a node is estimated as $E[\alpha]$ in [2][§]. When the newly deployed node shares sub-keys with existing neighboring nodes, $E[\alpha]$ is equivalent to the difference $r_2 - r_1$. While the age of node A is 0, the average age of existing neighboring nodes is $E[\alpha]$. Therefore, $p_{i,S-RKP}$ is defined using $E[\alpha]$ as follows:

$$\begin{aligned} p_{i,S-RKP} &= p_{(i,(E[\alpha]))} \\ &= \frac{\binom{P-\delta E[\alpha]}{i} \cdot \sum_{x=0}^{m-i} \binom{P-\delta E[\alpha]-i}{x} \binom{\delta E[\alpha]}{m-i-x} \binom{P-i-x}{m-i}}{\binom{P}{m}^2} \end{aligned} \tag{12}$$

POLISH: In a deterministic key-sharing scheme, POLISH has an advantage that the probability of establishing a secure link is 100%, because a sensor s_i has a polynomial $f(x, ID_i)$ and also shares the pairwise symmetric key $K^r_{i,j} = f(ID_j, ID_i)$ with s_j in the first round (round 1). After that the pairwise symmetric key of each link is updated, and hence the secure connectivity is 100% at every round.

$$p_{i,POLISH} = 1 \tag{13}$$

4.2. *Resiliency*

The degree of self-healing is measured by resiliency. Resiliency is estimated by the ratio of links that has not been compromised by the capture of nodes.

RPoK: We can measure the resiliency by the following analytical model in [7]. We obtain the analytical model by combining P_{RoK} [2] with the polynomial-based scheme [10], in order to dramatically improve resiliency (i.e., the ratio of compromised links). The ratio P_{RPoK} at generation j in RPoK is defined by

[§]How to derive $E[\alpha]$ is shown in [2]. For instance, $E[\alpha] = 2.5$ when $(P, m) = (10000, 250)$. We use this $E[\alpha]$ in this paper.

$$P_{RPoK} = 1 - \sum_{i=0}^{t} {}_{c \cdot E'_c}C_i \left(\frac{m}{P}\right)^i \left(1 - \frac{m}{P}\right)^{c \cdot E'_c - i} \tag{14}$$

Figure 6 shows a comparison of the analytical result (P_{RPoK}) with simulation result (R_S) ($t = 2$) [7] in RPoK. We found that the resiliency ($1 - R_{RPoK}$) $= 99.4\%$ by Figure 6 holds, where $E'_c = 3$ and $c = 10$. We see that the resiliency is much higher than other two schemes by Table II.

S-RKP: We can measure the resiliency by the following analytical model in [15]. The idea of modeling the RoK scheme is to replace the generation j by the constant value. We evaluate this scheme employing the modeling method of RoK, that is, we estimate a constant value and replace it by j. Then, the ratio $P_{S\text{-}RKP}$ in [15] is defined by

$$P_{S-RKP} = \sum_{i=1}^{m} \left(1 - \prod_{r=1}^{E''_c} \left(1 - \frac{m}{P + (r-1)\delta}\right)\right)^i \frac{p_{i,S-RKP}}{1 - p_{0,S-RKP}}, \tag{15}$$

where $E''_c = P/2\delta$. We can easily confirm that Equation (15) is the extended form of P_{RoK} by [7]. Figure 6 shows a comparison of the analytical result (P_{S-RKP}) with simulation result (R_S) [15] in S-RKP. We found that the resiliency ($1 - P_{S-RKP}$) $= 82.3\%$ by Figure 6 holds when $\delta = 100$. This means that S-RKP can achieve resiliency without using the security executions.

POLISH: Unlike the POSH scheme, a sensor in POLISH receives contributions from neighboring sensors, that is, a sensor receives w contributions. Note that the state transition of a sensor is the same as in the POSH scheme. In POLISH, it is necessary to consider the contributions

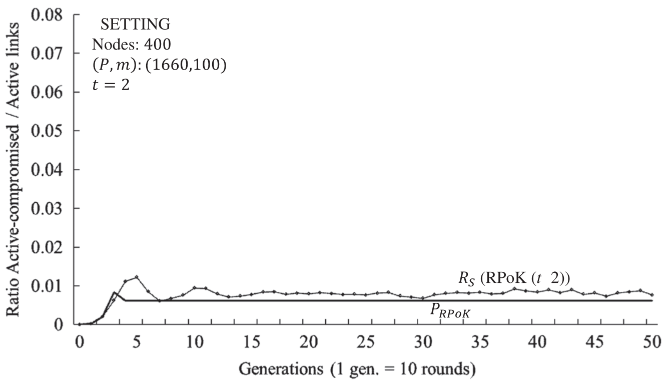

Figure 6. Polynomial-based self-healing scheme (RPoK): analytical results and simulation results against eager attackers.

Table II. Comparison of self-healing schemes for wireless sensor networks (WSNs).

	Self-healing WSNs			
	RoK[2]	RPoK[7]	S-RKP[15]	POLISH[16]
Key management by server	Necessary	Necessary	Necessary	no
Sever overhead	Hash	Hash + Poly	Hash	no
Sensor overhead	Hash	Hash + Poly	no	Hash + PRF
Round synchronization	Necessary	Necessary	no	Necessary
Secure connectivity (%)	99.8	99.8	99.2	100
Resiliency (%)	92.8	99.4	82.3	90.1

RPok, polynomial-based self-healing scheme; S-RKP, simple random key pre-distribution scheme with self-healing; POLISH, proactive co-operative link self-healing; PRF, Pseudo Random Function.

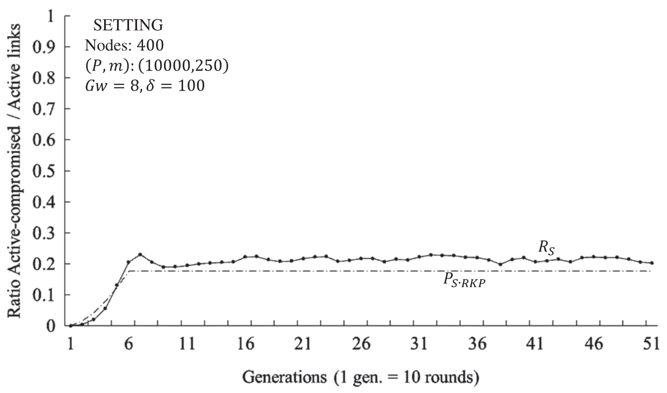

Figure 7. Simple random key pre-distribution scheme with self-healing (S-RKP): analytical results and simulation results against eager attackers.

from two-hop neighboring sensors. The contributions from neighboring sensors may be eavesdropped on by two-hop neighboring sensors. In this case, a green sensor is not self-healed even if it obtains a contribution from a green sensor. Let $\left(1-(1-p_{R^r})^{w-1}\right)$ be the probability that at least one sensor of two-hop neighboring sensors is red, that is, the probability that a green sensor's contribution is eavesdropped on by an adversary (i.e., red sensor), which is within the wireless communication range of the green sensor. To become a green sensor (from yellow), the yellow sensor needs to be linked with at least one green sensor among neighboring sensors, and also a red sensor must not be within the wireless communication range of that green sensor. Thus, the probability of a yellow sensor not becoming green can be expressed as follows:

$$Pr' = \sum_{i=0}^{w} \binom{m}{i} p_{G^r}^{i} (1-p_{G^r})^{w-i} \left(1-(1-p_{R^r})^{w-1}\right)^{i}, \tag{16}$$

where $p_{G^r} = \frac{|G^r|}{n-1}$, $p_{Y^r} = \frac{|Y^r|}{n-1}$ and $p_{R^r} = \frac{|R^r|}{n-1}$. The expected number of green sensors at round r is the same as in the POSH scheme as follows[¶]:

$$E[|G^{r+1}|] = |G^r| + (1-Pr')|Y^r| - |R^r| \tag{17}$$

To evaluate the link-healing rate of POLISH, we analyze the number of green links by evaluating the state of sensors in any round, that is, the number of G(G)G in Figure 5. The partial state transition diagram of a link is shown in Figure 5, in which only the transition required to analyze the number of green links is depicted. That is, we consider only the input and the output of G(G)G and G(R)G. Let $\alpha_1, \alpha_2, \beta, \gamma_1, \gamma_2$, and γ_3 be the number of link state transition (use not probability but a number.) and let $RL^r_{G(R)G} \subset RL^r$ be a set of the link state G(R)G. This figure shows that the expected number of green links in round r is

$$E[|GL^{r+1}|] = |GL^r| + \alpha_1 + \alpha_2 - \beta, \tag{18}$$

where $\alpha_1 = (1-(1-(1-p_{R^r})^{w-1})^2)|RL^r_{G(R)G}|$, $\alpha_2 = (1-Pr')|Y^r|p_{\alpha_2}$, and $\beta = |R^r|p_\beta$. α_1 is the number of green links between two green sensors changed from $RL^r_{G(R)G}$. This transition occurs if neither of the green sensors is linked with a red sensor. Let p_{α_2} be the probability that a sensor needs to be linked with at least one green sensor of the neighboring sensors, and also that a red sensor must not be within the wireless communication range of that green sensor. α_2 is the number of green links between two green sensors, changed from red links between a green sensor and a yellow sensor. Let p_β be the probability that at least one green sensor in GL^r is corrupted.

[¶]Because we assume that an adversary corrupts only the green sensor (i.e., INF-ADV in [13]), we can use not inequality but an equation.

Hence, β is the number of red links between two red sensors, or between a yellow sensor and a red sensor changed from GL^r, because an adversary corrupts only the green sensors and the number of adversaries is $|R^r|$ in any round.

The number of red links between two green sensors is estimated in Figure 5 as follows:

$$E\left[\left|RL^{r+1}_{G(R)G}\right|\right] = \left|RL^{r}_{G(R)G}\right| - \alpha_1 + \gamma_1 + \gamma_2 - \gamma_3, \tag{19}$$

where $\gamma_1 = (1 - Pr')|Y^r|p_{\gamma_1}$, $\gamma_2 = (1 - Pr')|Y^r|p_{\gamma_2}$, and $\gamma_3 = |R^r|p_{\gamma_3}$. Let p_{γ_1} be the probability that a sensor is linked with a green sensor, and also that a red sensor must not be within the wireless communication range of that green sensor. Let p_{γ_2} be the probability that a sensor is linked with a yellow sensor, which becomes green. Moreover, let p_{γ_3} be the probability that at least one green sensor in $RL^r_{G(R)G}$ is corrupted. The ratio of red links is denoted by P_{POLISH}.

Let μ be the ratio of $|GL^r|$ in a set of two neighboring green sensors, which are linked each other, that is, $\mu = \frac{|GL^r|}{|GL^r| + \left|RL^r_{G(R)G}\right|}$. We show the probability of p_{α_2}, p_β, p_{γ_1}, p_{γ_2}, and p_{γ_3} as follows:

$$\begin{aligned}
p_{\alpha_2} &= \sum_{i=0}^{w} \binom{w}{i} \left(p_{G^r}\,(1-p_{R^r})^{w-1}\right)^i \left(1 - p_{G^r}\,(1-p_{R^r})^{w-1}\right)^{w-i} i \\
p_\beta &= \sum_{i=0}^{w} \binom{w}{i} (p_{G^r}\mu)^i\,(1-p_{G^r}\mu)^{w-i}\, i \\
p_{\gamma_1} &= \sum_{i=0}^{w} \binom{w}{i} \left(p_{G^r}\left(1-(1-p_{R^r})^{w-1}\right)\right)^i \left(1 - p_{G^r}\left(1-(1-p_{R^r})^{w-1}\right)\right)^{w-i} i \\
p_{\gamma_2} &= \sum_{i=0}^{w} \binom{w}{i} \left((1-Pr')\,p_{Y^r}\right)^i \left(1-\left(1-Pr'\right)p_{Y^r}\right)^{w-i} i \\
p_{\gamma_3} &= \sum_{i=0}^{w} \binom{w}{i} (p_{G^r}(1-\mu))^i\,(1-p_{G^r}(1-\mu))^{w-i}\, i
\end{aligned} \tag{20}$$

Figure 8 shows a comparison of the analytical results and the simulation results. We found that the resiliency $(1 - P_{POLISH}) = 90.1\%$ by Figure 8 holds.

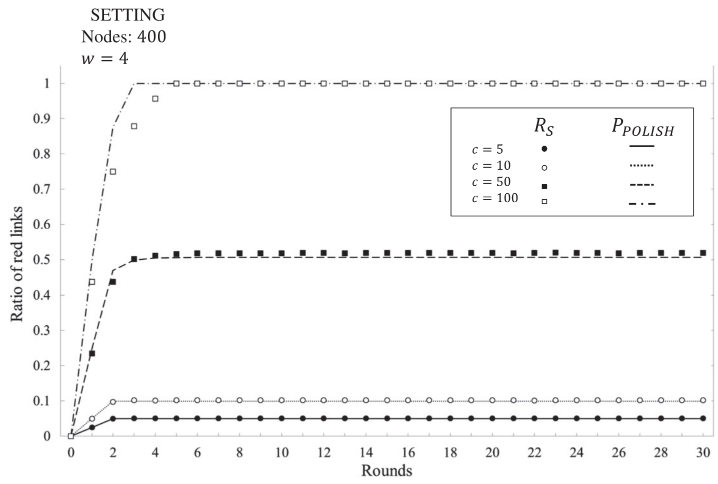

Figure 8. Proactive co-operative link self-healing (POLISH): analytical results and simulation results against eager attackers.

5. SECURITY EVALUATION BY SIMULATION

We evaluate the ratio of links compromised by eager attackers to show the improvement of resiliency in our schemes. For ease of exposition and without loss of generality, we assume that the rounds of node compromising have the same duration. The ratio of compromised links is defined as

$$R_S = \frac{\text{active-compromised links}}{\text{active links}} \tag{21}$$

We follow the RoK scheme [2] regarding parameters and network. To simplify the security analysis, we model the network as a grid of sensors of size $n = 400$ (20×20). We assume that the number of neighbors of each sensor is constant and equal to four. This type of network topology is a mesh network. We also assume that the network topology does not change over time. The simulations are implemented in C. We set the parameters of RPoK and S-RKP to establish 'almost certain' shared-key connectivity.

5.1. Polynomial-based self-healing scheme and simple random key pre-distribution scheme with self-healing

Polynomial-based self-healing scheme and S-RKP are probabilistic key-sharing schemes with self-healing.

Simulation setup:

- *Parameters*: The maximum life of a sensor is set to 100 rounds (i.e., $Gw = 10$ generations), which is the same parameter as [2]. In RPoK, P and m are decided not only by the degree t but also by the secure connectivity. When we consider the relation between P and m under the condition $p_{i,RPoK}$, $p_{i,S\text{-}RKP} \geqslant 0.998$, we can set $(P, m) = (1660, 100)$ for $t = 2$ and $(P, m) = (1158, 83)$ for $t = 3$ in RPoK and $(P, m) = (10000, 250)$ in S-RKP, which are required to establish 'almost certain' shared-key connectivity. We also set $\delta = 100$ in the simulation of S-RKP. Note that the evaluation changing δ is conducted in Section 6. All the simulations were repeated 25 times, and the results report the average values.
- *Network*: We assume that one generation consists of 10 rounds ($r = 10$) and that the attacker corrupts one active sensor at each round ($c = 1$). At each generation, expired nodes are replaced with new ones, configured with fresh keys. The new nodes establish secure links with their four neighbors using session keys. More importantly, a round synchronization is not necessary for S-RKP, although it is necessary for RPoK.

Simulation details: We evaluate the security of RPoK and S-RKP by the number of links that becomes indirectly corrupted when the nodes are compromised. A link, between nodes A and B, is said to be indirectly corrupted when neither A nor B has been corrupted, but when the adversary has collected all the sub-keys that A and B have in common. These sub-keys have been collected by compromising other nodes.

At the beginning of round 0, n nodes are deployed. We simulated nodes, expiration by assigning to each node a random expiration date, chosen according to a Gaussian distribution with mean $Gw/2$ and with standard deviation $Gw/6$ [2]. Thus, sub-keys have limited lifetimes (i.e., the mean life is five generations (50 rounds)) and are refreshed periodically.

The attacker may create a table of keys that belongs to various rounds. He corrupts one active node at each round and updates such a table. He then uses this table to corrupt links. We counted, at each generation, the number of compromised links and computed the ratio R_S. An attacker does not capture a node, which has already been corrupted to deal with the most serious situation, because he has all secret information of corrupted sensors.

Simulation results: Figures 6 and 7 display the ratio R_S of RPoK and S-RKP against an eager attacker, respectively. The R_S of RPoK and S-RKP is suppressed to about 0.0081 ($t = 2$) and 0.214, respectively. These results show that our schemes have self-healing property and that they also hold both forward and backward security. The fluctuation of R_S indicates probabilistic forward and backward security. We found that our analytical results well matched the simulation results of RPoK and S-RKP.

5.2. *Proactive co-operative link self-healing*

Proactive co-operative link self-healing is a deterministic key-sharing scheme with self-healing. We evaluate the ratio of red links against eager attackers to show the resiliency of POLISH. For ease of exposition and without loss of generality, we assume that each round when sensors are compromised has the same duration and is synchronized.

Simulation setup: All the simulations are repeated 25 times, and the results show the average values.

Simulation details: We evaluate the security of POLISH by the number of red links when an adversary can compromise c sensors from the set G^r in any round. At the first round (round 1), n green sensors are deployed. An eager attacker keeps compromising sensors at constant rate from the deployment of the first round of sensors to the end of the network. We then counted, in each round, the number of red links and computed the ratio. With the eager attacker, we ran the simulation until (1) the WSN has no more green sensors or (2) $|R^r|$ reaches a steady state.

Simulation results: Figure 8 displays the ratio of red links against eager attackers. The ratio of red links is suppressed to 5.1% with $c = 5$, 10% with $c = 10$, 52% with $c = 50$, and 100% with $c = 100$, depicted in Figure 8. We found that our analytical results well matched the simulation results of POLISH.

6. DISCUSSION

6.1. *Analysis of simple random key pre-distribution scheme with self-healing*

There is typically a tradeoff between secure connectivity and resiliency against node-capture attacks in the probabilistic key-sharing schemes. It is desirable that both values are high. Fortunately, the tradeoff is hardly appeared in S-RKP. δ can be set in such a way that it gives the maximum resiliency without spoiling secure connectivity.

Figure 9 shows the changes of secure connectivity and resiliency when changing δ. The resiliency is derived using $1 - P_{S\text{-}RKP}$ and then has the maximum value as described in this figure. For example, the δ, which makes the resiliency maximum is 180 in Figure 9. Figure 10 represents δ, which makes the resiliency maximum when $2 \leqslant Gw \leqslant 20$. Note that Gw means the maximum lifespan of a sensor. We can determine the renewal ratio δ according to Gw from the viewpoint of self-healing.

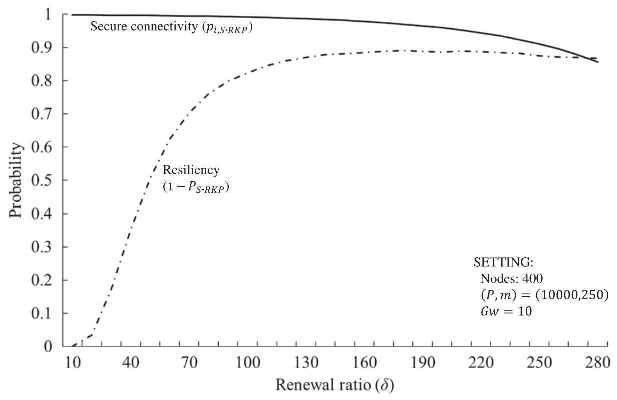

Figure 9. Analytical results of resiliency and secure connectivity by changing δ ($Gw = 10$).

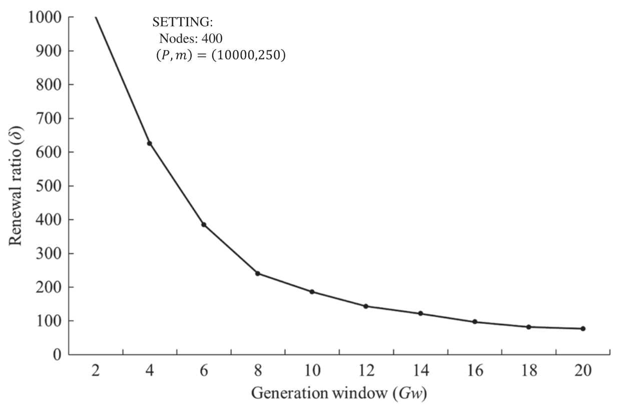

Figure 10. Renewal ratio (δ) for each Gw with the maximum resiliency.

6.2. *Comparison simple random key pre-distribution scheme with self-healing with RoK*

The RoK scheme was the first resilient multiphase WSN scheme with self-healing property and was the most efficient. On the other hand, S-RKP is an efficient and resilient multiphase WSN scheme with self-healing property. The goal of this section is to evaluate secure connectivity and resiliency in S-RKP and compare them with the RoK scheme. The 'almost certain' shared-key connectivity is assumed. We compare S-RKP with the RoK scheme, following a similar simulation procedure as in [2]. When the length of each ring is m, the total number of sub-keys of RoK is just $2m$. Thus, when the size of the key ring of S-RKP is m, we set $\frac{m}{2}$ as the length of a ring for the RoK scheme, from the standpoint of fairness. More concretely, $m = 250$ for S-RKP and $m = 125$ for the RoK scheme are set in this comparison. Also, we employ $Gw = 10$, which is the same as the parameter in [2].

At first, we compare the secure connectivity of S-RKP with that of the RoK scheme. We set P so that the secure connectivity of S-RKP ($\delta = 0$) becomes the same as that of the RoK scheme. The secure connectivity of the RoK scheme and S-RKP ($\delta = 0$) is the same (i.e., $p_{i,S\text{-}RKP} = 99.8\%$). Actually, we can derive P from Equation (1) so that the secure connectivity of the RoK scheme is $p_i = 99.8\%$, that is, we obtain the adjusted parameters $(P, m) = (2562, 125)$. While the secure connectivity is constant ($p_i = 99.8\%$) in the RoK scheme, the secure connectivity changes by renewal ratio δ in S-RKP. Hence, as δ becomes larger, secure connectivity decreases slightly smaller in S-RKP. Figure 11 shows the transition of secure connectivity of the RoK scheme and of S-RKP and also shows that secure connectivity is maintained highly in S-RKP.

Remark: In the RoK scheme, each node updates its key ring. Thanks to this mechanism, the secure connectivity of the RoK scheme does not decrease, even if the updating ratio of the key pool changes. On the contrary, each node does not update its key ring in S-RKP.

Then, we compare the resiliency of S-RKP with that of the RoK scheme. More precisely, we compare $P_{S\text{-}RKP}$ with P_{RoK} by the same parameter as in the case of the secure connectivity. Figure 12 shows the ratio of compromised links of the RoK scheme and S-RKP ($\delta = 100$), where P_{RoK} denotes the ratio of compromised links against eager attackers. Concretely, $p_{i,S\text{-}RKP} = 99.2\%$ and $P_{S\text{-}RKP} = 0.177$ hold when $\delta = 100$. Note that $p_{i,S\text{-}RKP}$ is decreased from 0.998 to 0.992 when δ increases from 0 to 100. Actually, the stable point of the ratio of compromised links varies by changing δ in S-RKP, and it is thus necessary to evaluate the ratio of compromised links by changing δ. Figure 13 shows the transition of the stable point of the ratio of compromised links by changing δ in Equation (15). When $\delta \geqslant 87$, the ratio of compromised links of S-RKP becomes lower than that of the RoK scheme. Of course, $\delta < P$ holds. Furthermore, $P_{S\text{-}RKP}$ has a minimum value in $\delta \geqslant 87$. S-RKP achieves resilient multiphase WSNs without even lightweight operations such as a hash function, although secure connectivity decreases a little.

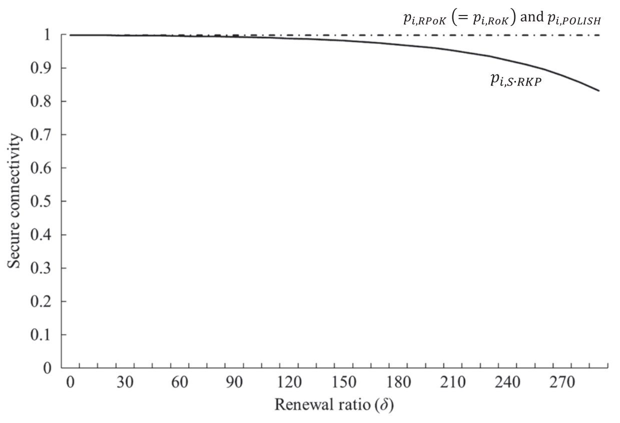

Figure 11. Analytical comparison of secure connectivity by changing δ. RPoK, polynomial-based self-healing scheme; POLISH, proactive co-operative link self-healing; S-RKP, simple random key pre-distribution scheme with self-healing.

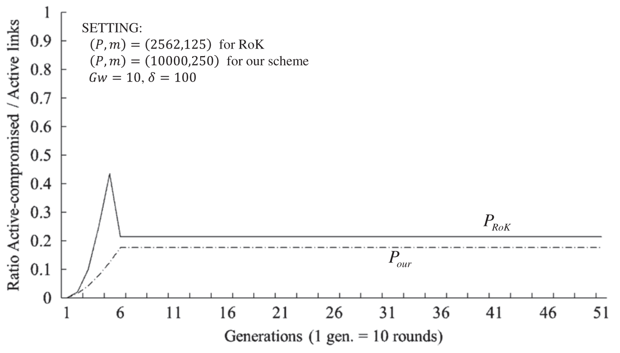

Figure 12. Analytical comparison of the ratio of compromised links.

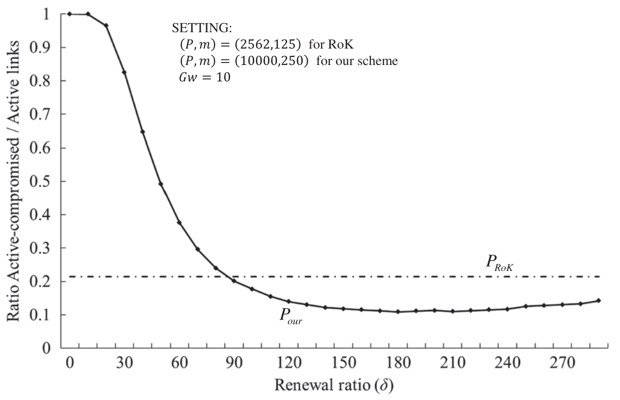

Figure 13. Analytical comparison of the ratio of compromised links by changing δ.

6.3. *Comparison of our schemes*

Table II shows the comparison of three self-healing schemes for WSNs from the viewpoint of security computation and procedure overhead that sensor/server execute. Figure 14 also displays the resiliency of RPoK, S-RKP, and POLISH against eager attackers, whose graphs are the same as the graphs in Figures 6–8. We set the parameters of RPoK and S-RKP to establish 'almost certain' shared-key connectivity (Note that secure connectivity of POLISH is 100%.). Secure connectivity and resiliency are the results of each analytical evaluation. Furthermore, the number of attackers is the same (i.e., $c = 10$) at each generation (note that we need to consider 'round' as 'generation' in POLISH.).

Polynomial-based self-healing scheme can dramatically increase resiliency, and hence it is suitable for situations that require higher resiliency such as a more hostile area, compared with the other two schemes. On the other hand, S-RKP can enhance the RKP with self-healing property, without changing the functions of the sensors. Thus, S-RKP can increase resiliency under realistic assumptions because a sensor does not have the functions of security executions and round synchronization. This means that S-RKP is suitable for WSNs that use resource-poor sensors although resiliency is somewhat low, compared with the other two schemes. Note that secure connectivity of RPoK and S-RKP is not 100% because they are probabilistic key-sharing schemes.

Proactive co-operative link self-healing is suitable for the situations where 100% secure connectivity is required and the size of memory is quite efficient in POLISH, because it is a deterministic key-sharing scheme. Also, POLISH can keep higher resiliency without the help of a server, where the sensor operates independently. Hence, POLISH is suitable for WSNs where the key management is not necessary by a server.

6.4. *Computational, communication, and memory costs*

Table III shows the computational cost, the communication cost, and the size of memory in self-healing schemes for WSNs. Let $\mathcal{M}$ and $\mathcal{R}$ be the multiple operation over a finite field $\mathbb{F}_q$ and the PRNG operation, respectively and also let η be the length of each key ring. Let $|q|$ be the size of the sub-keys, contribution, ID, the output of hashing, and the coefficient of a polynomial.

The computational cost of each sensor in a round is discussed here. The computational cost of RPoK is a little larger than the one of the RoK. As for the computational cost of link establishment,

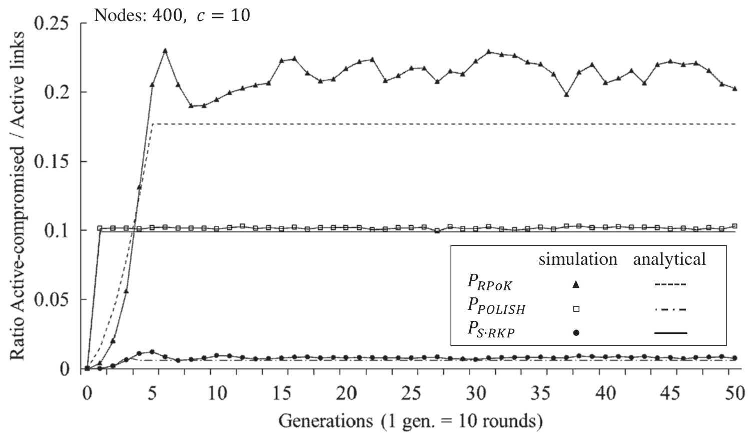

Figure 14. Comparison of results of polynomial-based self-healing scheme (RPoK), simple random key pre-distribution scheme with self-healing (S-RKP) and proactive co-operative link self-healing (POLISH).

Table III. Computational, communication, and memory costs of each sensor.

	Self-healing wireless sensor networks			
	RoK[2]	RPoK[7]	S-RKP[15]	POLISH[16]
Computational cost	$(m+1)H$	$(2m+1)H + \frac{m^2}{4}F + \frac{mt+m+2t}{2}\mathcal{M}$	–	$(w+1)H + w\mathcal{R}$
Communication cost	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$2w\|q\| + \mathcal{D}$
Memory cost	$2\eta\|q\|$	$(t+3)\eta\|q\|$	$\eta\|q\|$	$(t+3)\|q\|$

RPok, polynomial-based self-healing scheme; S-RKP, simple random key pre-distribution scheme with self-healing; POLISH, proactive co-operative link self-healing.

that for RPoK is $H + \frac{m^2}{4}F + t\mathcal{M}$, while that for RoK is just H. Note that the computational cost of F is much lower than H. As for the computational cost of key update, that for RPoK is $2mH + \frac{m(t+1)}{2}M$, while that for RoK is mH. The computational cost of S-RKP is the same as mere RKP, hence it is so small that it can be ignored. On the other hand, the computational cost of POLISH is $w\mathcal{R}+(w+1)H$. Note that it is required for s_i to assign the value of ID_j to a polynomial $f(x, ID_i)$ to share the pairwise symmetric key only in the first round (round 1).

The communication costs of each sensor in a round are as follows: The communication cost of RPoK is the same as that of the RoK scheme, because the communication in both schemes is required in only the neighbor discovery procedure of establishing a secure link, denoted by $\mathcal{D}$. The communication cost of POLISH is $2w|q|$, which includes the contributions of transmission and reception. Note that s_i needs to obtain IDs from w neighboring sensors in round 1.

The size of memory of each sensor is discussed here. Especially, it is evaluated by the size of keys and key materials. The total size of two key rings of RoK is just 2η, while the size of two key rings and coefficients in RPoK is η, η and $\eta(t+1)$, respectively, that is in total $\eta(t+3)$. In order to setup data of a sensor at the first round in POLISH, s_i requires a seed, ID and, the coefficient of a polynomial, that is, the size of memory on a sensor requires $(t+3)|q|$ in total. After key establishment, s_i deletes all the coefficients of a polynomial, but w pairwise symmetric keys whose sizes are $w|q|$ are generated. Thus, the amount of memory on a sensor can save $(t+1-w)|q|$. This means that s_i can keep the contributions of transmission and reception if $(t+1) \geqslant 3w$. Therefore, POLISH is efficient and is suitable for WSNs, which constitute sensors with both limited memory and limited computational power.

Remark: The computational cost, the communication cost, and the size of memory of S-RKP is the same as those of the RKP scheme, because the processing of each sensor of S-RKP is the same as in the RKP schemes.

7. CONCLUSION

We summarize the three kinds of self-healing schemes for WSNs. RPoK is a scheme, which emphasizes security (resiliency), and it is suitable in situations that require higher resiliency such as a very hostile area. S-RKP is a scheme, which emphasizes efficiency of a sensor, and it is suitable for WSNs, which use resource-poor sensors. They are probabilistic key-sharing schemes, and they can attach self-healing property to existing RKP schemes. On the other hand, POLISH is a scheme, which emphasizes the sensor operations without the help of a server. Hence, POLISH is suitable for WSNs where the key management is not necessary by a server, and it is also suitable for the situations where 100% secure connectivity is required because it is a deterministic key-sharing scheme.

ACKNOWLEDGEMENTS

This study is partly supported by Grant-in-Aid for Scientific Research (A)(i21240001), Grant-in-Aid for Young Scientists (B) (25730083) and CREST, JST.

REFERENCES

1. Eschenauer L, Gligor VD. A key-management scheme for distributed sensor networks. *CCS 2002*, Scottsdale, Arizona, USA; 41–47.
2. Castelluccia C, Spognardi A. Rok: A robust key pre-distribution protocol for multi-phase wireless sensor networks. *SecureComm 2007*, Nice, France; 351–360.
3. Yilmaz OZ, Levi A, Savas E. Multiphase deployment models for fast self healing in wireless sensor networks. *SECRYPT 2008*, Porto, Portugal; 136–144.
4. Kalkan K, Yilmaz S, Yilmaz OZ, Levi A. A highly resilient and zone-based key predistribution protocol for multiphase wireless sensor networks. *Q2SWinet 2009*, The Canary Islands, Spain; 29–36.
5. Ergun M, Levi A, Savas E. Increasing resiliency in multi-phase wireless sensor networks: generationwise key predistribution approach. *The Computer Journal* 2011; **54**(4):602–616.

6. Tian W, Han S, Parvin S, Dillon TS. A key management protocol for multiphase hierarchical wireless sensor networks. *EUC 2010*, Hong Kong, China; 617–623.
7. Ito H, Miyaji A, Omote K. RPoK: a strongly resilient polynomial-based random key pre-distribution scheme for multiphase wireless sensor networks. *GLOBECOM 2010*, Miami, Florida, USA; 1–5.
8. Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks. *SP 2003*, Oakland, CA, USA; 197–213.
9. Blundo C, Santis AD, Herzberg A, Kutten S, Vaccaro U, Yung M. Perfectly-secure key distribution for dynamic conferences. *CRYPTO' 1992,* LNCS, 740, Santa Barbara, CA, USA, 1993; 471–486.
10. Liu D, Ning P, Li R. Establishing pairwise keys in distributed sensor networks. *ACM Transactions on Information and System Security* 2005; **8**(1):41–77.
11. Du W, Deng J, Han YS, Chen S, Varshney PK. A key management scheme for wireless sensor networks using deployment knowledge. *INFOCOM 2004*, Hong Kong, China; 586–597.
12. Da AK. A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks. *International Journal of Information Security* 2012; **11**(3):189–211.
13. Pietro RD, Ma D, Soriente C, Tsudik G. POSH: proactive co-operative self-healing in unattended wireless sensor networks. *SRDS*, Napoli, Italy, 2008; 185–194.
14. Ma D, Tsudik G. Distributed self-healing. *SSS*, Detroit, Michigan, USA, 2008; 47–62.
15. Miyaji A, Omote K. How to build random key pre-distribution schemes with self-healing for multiphase WSNs. *AINA*, Barcelona, Spain, 2013; 205–212.
16. Iida T, Miyaji A, Omote K. Proactive co-operative link self-healing for wireless sensor networks. *SSS 2011*, Grenoble, France; 253–267.
17. Pietro RD, Oligeri G, Soriente C, Tsudik G. Intrusion-resilience in mobile unattended WSNs. *INFOCOM*, San Diego, CA, USA, 2010; 2303–2311.