Ad Hoc Networks 71 (2018) 68–77

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Key establishment scheme for wireless sensor networks based on polynomial and random key predistribution scheme

Jianmin Zhang*, Hua Li, Jian Li

School of Computer, Henan Institute of Engineering, Zhengzhou 451191, China

ARTICLE INFO

Article history:
Received 5 April 2017
Revised 8 November 2017
Accepted 20 December 2017
Available online 30 December 2017

Keywords:
Wireless sensor network
Security
Key predistribution
Polynomial-based key predistribution

ABSTRACT

Establishing communication keys for pairs of neighbouring sensor nodes is the foundation of the security in wireless sensor networks (WSNs). However, due to the resource constraints on nodes, this task is challenging for the constrained memory, energy, and computational capabilities of sensor nodes. This paper proposes a novel key predistribution scheme based on the polynomial pool-based key predistribution scheme and random key predistribution. In the proposed scheme, parts of the preloaded information in each sensor node are the polynomial shares and the rest of the preloaded information are the keys generated by the polynomial shares preloaded in the sensor nodes. Performance analyses and comparisons with other schemes are performed in this paper. The comparison of security results confirm that the proposed scheme has better resilience against node compromising attacks when compared to previous schemes.

1. Introduction

Wireless sensor networks (WSNs) usually include battery-powered sensor nodes that are deployed in a designed area to sense and collect information [1]. That information is transmitted by sensor nodes to the sink node where it is aggregated [2]. The applications of WSNs include environmental monitoring, health care, battlefield targeting and surveillance, and disaster relief networks [3,4]. However, WSNs are usually subject to many security threats and attacks such as impersonation, intentionally providing false information, eavesdropping, data modification, and sensor node capture attacks.

Confidentiality, authenticity, availability, and integrity are typical security goals of WSNs [5]. As the basic requirement for providing security functionality, key management is an important fundamental security service that enables sensor nodes to securely communicate with each other using cryptographic techniques [6]. The prime problem in key management is the establishment of a secure key shared between two communicating sensor nodes [7]. Meanwhile, inherent constraints of computational power, memory capability, and bandwidth make the direct use of traditional pairwise key establishment algorithms (such as public key cryptography and key distribution centre (KDC)) in WSNs impossible [8]. In recent years, many researchers have studied key distribution in WSNs, and several key distribution schemes have been proposed. In these schemes, the key predistribution schemes are the most suitable for WSNs. In these key predistribution schemes, key generation materials are preloaded to sensor nodes before network deployment and the two nodes can use these materials to establish communication keys after the network is deployed [9].

In this paper, we propose a new key predistribution mechanism based on the polynomial pool key predistribution scheme and the random probabilistic key predistribution scheme. In the proposed scheme, the preloaded key generation materials in sensor nodes are divided into two parts. One part contains the polynomials randomly selected from a polynomial pool. The other part is loaded with the keys calculated by the polynomial shares preloaded in the sensor nodes. This proposed scheme guarantees that an attacker can't obtain any information from the polynomials in the uncompromised nodes from the compromised sensor nodes. The contributions and findings of this study provide a new approach for further exploiting the polynomial shares preloaded in the sensor nodes, as one polynomial share preloaded in the sensor nodes can play two roles when establishing communication keys between sensor nodes. The main benefit of the method is that this scheme offers further trade-offs between the security against node captures and the probability of establishing pairwise keys directly between neighbouring nodes given a certain memory constraint.

The organization of the rest of this paper is as follows. In Section 2, the related works are examined. Section 3 describes the overview of the polynomial-based key predistribution scheme. Section 4 presents the proposed scheme. In Section 5, the proposed scheme is evaluated and compared with previous schemes. Finally, Section 6 presents the conclusions.

* Corresponding author.
E-mail address: zjm7008@163.com (J. Zhang).

https://doi.org/10.1016/j.adhoc.2017.12.006
1570-8705/© 2017 Published by Elsevier B.V.

2. Related works

There are a number of key predistribution schemes that are mainly based on two types of network structures:the distributed WSN and the hierarchical network. In the distributed WSN, all nodes have the same capabilities, the roles of all sensors are similar, and communications may occur among any pair of neighbours. In a hierarchical network [10–12], sensor nodes are grouped together to form a cluster. In each cluster, a single node is selected as the cluster head (CH) while the remaining nodes become member nodes. Member nodes sense environmental data and transmit it to the CH. The responsibility of the CH is to aggregate that data and transmit it to the sink node. Although cluster-based networks are efficient in terms of scalability and energy, the scheme consumes additional resources due to the network's need to periodically reorganize the cluster. In this paper, we only consider key management for the distributed WSN.

Eschensuer and Gligor [13] proposed a Random Key Predistribution (RKP) scheme for WSNs based on probabilistic key sharing among the nodes of a random graph. This scheme consists of three phases: the key predistribution phase, the shared-key discovery phase and the path key establishment phase. The key predistribution phase is performed before network deployment. In this phase, the Key Distributions Servers (KDS) generate a large key pool and each key in the key pool has a distinct key identification. Each sensor node randomly picks a set of keys from the key pool such that any two sensor nodes have a certain probability of sharing at least one common key. In the shared key discovery phase, nodes can discover the shared keys it has with its neighbours by exchanging the key identifiers stored in their storage. If two nodes have no shared keys, they will establish communication by path key establishment. In this key predistribution approach, the number of preloaded keys in each node and the key pool size are chosen in such a way that the intersection of two key rings has a high probability of not being empty. This basic approach is energy efficient, but it requires a large memory space to store the key rings. Moreover, if the network nodes are progressively corrupted, an attacker may discover a large part of the global key pool or the whole global key pool. Hence, a great number of links will be compromised. In this scheme, there is a tradeoff between connectivity and security. With a given sized key pool, more memory will be used to preload the keys into each sensor node. This leads to higher connectivity and worse security. Chan et al. further extended this idea and developed the q-compromised scheme [14]. The q-composited scheme requires that two nodes must share at least q $(q > 1)$ common keys to establish a secure communication. In this scheme, the communication key is generated by performing a hash function on the concatenation of all common keys. The analyses show that this scheme has better performance against node capture attacks than the RKP scheme when there are few compromised nodes. Once the number of the compromised nodes become larger, the security performance of the q-composite scheme is worse than that of the RKP scheme.

In [15], the authors proposed a new key predistribution scheme that combines the original polynomial-based key distribution scheme [16] and the RKP scheme. This scheme uses a polynomial pool instead of a key pool in the RKP and q-composite schemes. The sensor nodes are the preloaded polynomial shares as computed in [16]. If two nodes have polynomial shares that are computed from the same polynomial, they can generate a communication key based on the common polynomial. In [17], Du et al. developed a similar scheme in which they use the key spaces pool generated by matrices instead of the polynomial pool as in [15]. These schemes have been further studied [18–20]. The analyses show that these schemes have a threshold value, which implies that the compromised nodes do not leak the communication key between uncompromised nodes when the number of compromised nodes is less than the threshold. However, once the number of the compromised nodes is larger than the threshold, all communication keys between uncompromised nodes can be quickly calculated by the adversary. Another drawback of these schemes is that their computation overhead is much greater than that of the RKP and q-composite schemes.

Liu et al. [21] proposed several schemes based on deployment knowledge. These schemes take advantage of the prior knowledge about deployment and reduce the number of unnecessary key spaces carried by each sensor node. And these schemes can save storage while maintaining a high level of security. Du et al. [22] proposed a scheme in which the sensor nodes are deployed in group, nodes in the same group have higher probability of being deployed close to each other. In this scheme the original key pool is divides into many smaller pools and each of which is associated to different sensor node group. With this method, sensor node can avoid unnecessary key assignments and then save the storage. The group-based deployment schemes was further explored in [23–25]. These schemes using the deployment knowledge gain substantial improvement against the node capture attacks over exiting schemes that do not use deployment knowledge, but in most case we can't get the pre-deployment knowledge.

Camptepe and Yener [26] proposed key predistribution schemes based on the combinatorial design theory. Combinatorial design theory is the part of combinatorial mathematics that addresses the existence and construction of systems of finite sets whose intersections have specified numerical properties. In fact, the combinatorial design-based key predistribution scheme is a deterministic variant of the Eschensuer and Gligor scheme in which the key ring of each sensor node is deterministically designed. In particular, in [26], a mathematical structure called balanced incomplete block design (BIBD) is used to construct the key rings. The analysis and computational comparison to the randomized methods show that the combinatorial approach has two clear advantages. First, it increases the probability that a pair of sensor nodes shares a key. Second, it decreases the key-path length while providing the scalability of hybrid approaches. However, BIBD has its own constraint in that the number of sensor nodes must be of prime power. This significantly limits the use of this scheme. Thus, a hybrid scheme is proposed as a complement. The idea behind the hybrid design is that if the number of sensor nodes is not of prime power, the BIBD is constructed according to the closest prime power to the number of sensor nodes.

Other important key predistribution works proposed multi-phase WSNs [27–30]. In these multi-phase WSNs, the average lifespan of each node is assumed to be much shorter than the operating lifespan of the overall networks, and new sensor nodes will be periodically added to the networks [31]. If the schemes in [7–20] are directly used in multi-phase WSNs, the number of compromised links will constantly increase until all the links are compromised. In [26], the authors proposed a new predistribution scheme adapted to multi-phase WSNs. In the proposed scheme, the predistributed keys have limited lifespans and are periodically refreshed. Analysis results demonstrate that this scheme outperforms the RKP scheme used in multi-phase WSNs.

3. Overview of the polynomial-based key predistribution scheme

In this section, we briefly review the original polynomial-based key predistribution scheme [16] and the improved polynomial-based key predistribution scheme [15] which is suitable for use in WSNs.

In [10], the key setup server randomly generates some bivariate t-degree polynomials with the coefficients $f(x, y) = \sum_{0\le i,j\le t}^{t} a_{ij}x^iy^j$,

Table 1
Notations.

Notation	Description
N	The number of sensor nodes in the network
W	The storage per node in number of keys
ID_u	The identification of sensor node u
s	The number of keys preloaded in each sensor
m	The number of polynomials in the polynomial pool
t	The degree of the polynomials
Id_p	The identification of polynomial p
g	The number of polynomials selected for each node
H	One-way hash function
$\|\|$	Concatenation operation
$\overline{E}$	The complement of event E

where $a_{ij}=a_i$, over a finite field F_q, and q is a prime number large enough to accommodate a cryptographic key. Here, the polynomials need to have the property $f(x, y)=f(y, x)$. Each sensor node is assumed to have a unique *ID*. For each sensor node u, the setup server computes a polynomial share of $f(x,y)$ that is $f(u,y)$. The polynomial share $f(u,y)$ is loaded into sensor node u before node u is deployed. The polynomial shares are the coefficients b^i of x^i in the univariate t-degree polynomials $f(u, y)$. When the communication key is established, node u computes its key $f(u,v)$ by evaluating $f(u,y)$ at point v and node v computes the same key $f(v, u)=f(u, v)$ by evaluating $f(v,y)$ at point u.

In this scheme, each sensor node u needs $(t+1)$ log q storage space to store a t-degree polynomial $f(u,y)$. To establish a pairwise key, both sensor nodes need to evaluate the polynomial at the ID of the other sensor node. The advantage of this scheme is that any two neighbouring nodes can establish a communication key. The security proof for the Blundo scheme [10] ensures that this scheme is unconditionally secure if there are no more than t compromised sensor nodes. However, once the number of compromised nodes reaches t, all communication keys between uncompromised nodes will be computed by the adversary.

As the original polynomial-based key predistribution scheme can only be immune to no more than t compromised nodes (where the value of t is limited by the storage space and the computational capability of sensor nodes), it is unsuitable for sensor networks since it is more likely that an adversary compromises more than t sensor nodes and then the entire network. The authors in [9] proposed a key predistribution scheme based on the combination of the key pool idea used in [7] and the polynomial-based key predistribution from [10]. In this scheme, the key pool in [7] is replaced by the polynomial pool. For each sensor node, some polynomials are randomly chosen from the polynomial pool and their polynomial shares are assigned to sensor nodes.

4. The proposed scheme

4.1. Preliminaries

Here, we present definitions and notations that will be used in the rest of the paper.

1. Definitions

B-communication-keys: The communication keys between neighbouring nodes are computed using polynomials shares as in [15].

R-communication-keys: The communication keys between two neighbouring nodes are computed using the preloaded keys as in [13] or by using the pre-loaded key in one node and the polynomial shares in another node.

2. Notations

The notations used in rest of the paper are listed in Table 1.

4.2. Key predistribution scheme

As with the schemes in [13,15], the three phases in this proposed scheme are the Initialization Phase, the Direct Communication Key Establishment Path and the Path Key Establishment Phase. Because the last phase is not different from the schemes in [13,15], we will only discuss the first two phases.

4.2.1. Initialization phase

This initialization phase was performed by a Key Distributions Server (KDS) before network deployment. It includes the following three steps.

Step 1 (***Polynomial Share Predistribution for all Sensors***): Initially, the KDS generates a polynomial pool that contains m random bivariate t-degree polynomials over the finite field $F_{q,}$ and each polynomial has a unique identification Id_p. For each sensor u, the KDS randomly selects g polynomials from the polynomial pool. It then assigns their polynomial shares and the identification Id_p of the polynomial to the sensor node.

Step 2 (***Key Pool Generation***)**:** The KDS generates a key pool by using the polynomial shares in all sensor nodes. For example, by using the polynomial shares in sensor u, the KDS generates t keys $K=H(b_i||i||ID_u)$, where $0\le i\le t$ and b_i is the share of the polynomial. Each key in the key pool is identified by a 3-tuple (Id_p, i, ID_u).

Step 3 (***Key Predistribution for all Sensors***): For every sensor node, the KDS randomly picks s keys from the key pool. It stores these s keys and their corresponding identifications to the sensor node. Here, the tuple component Id_p in the identifications of all picked s keys should all be different within the same node.

An example of the initial phase is illustrated in Fig. 1.

4.2.2. Direct communication key establishment phase

Once the sensor nodes are deployed in the target field, every node tries to generate communication keys with any nodes in its communication range. To discover whether two neighbouring nodes can establish a common key, every node broadcasts the identifications of the g selected polynomials, the identification of the s preload keys and their identifications to its neighbours.

To clearly describe the key generation method, here we assume that sensor nodes u and v are neighbours and every node receives the broadcasting message sent from the other node. These two neighbouring nodes can calculate the communication key through the following method. According to the identifications of the polynomials and keys in the two nodes, three cases need to be considered.

Case 1: Nodes u and v have the same Id_p of the polynomial. In that case, these two nodes have the common polynomial selected from the polynomial pool and they can compute their communication keys as in [15].

Case 2: Sensor nodes u and v have the same identifications of keys. In that case, these two sensor nodes have common preloaded keys and can compute their communication keys as in [13].

Case 3: The identification Id_p of the predistributed polynomial in one node is the same as the tuple component Id_p in the identification of predistributed keys in another node. In that case, the preloaded keys in one node are generated using the shares of polynomials selected for another node.

Without losing generality, we suppose that the identification Id_p in node u is the same as the tuple component Id_p in the identification of one predistributed key K (Id_p, i, ID_w) in node v. The sensor node u can compute the key $K_{uv}=h(\mathrm{b_i}||i||ID_w)$ as the communication key. Sensor node v can directly use K as the communication key. The predistributed K in the sensor node v is $h(\mathrm{b_i}||i||ID_w)$.

In the above three cases, after the communication keys are generated, the two neighbouring nodes use the communication key

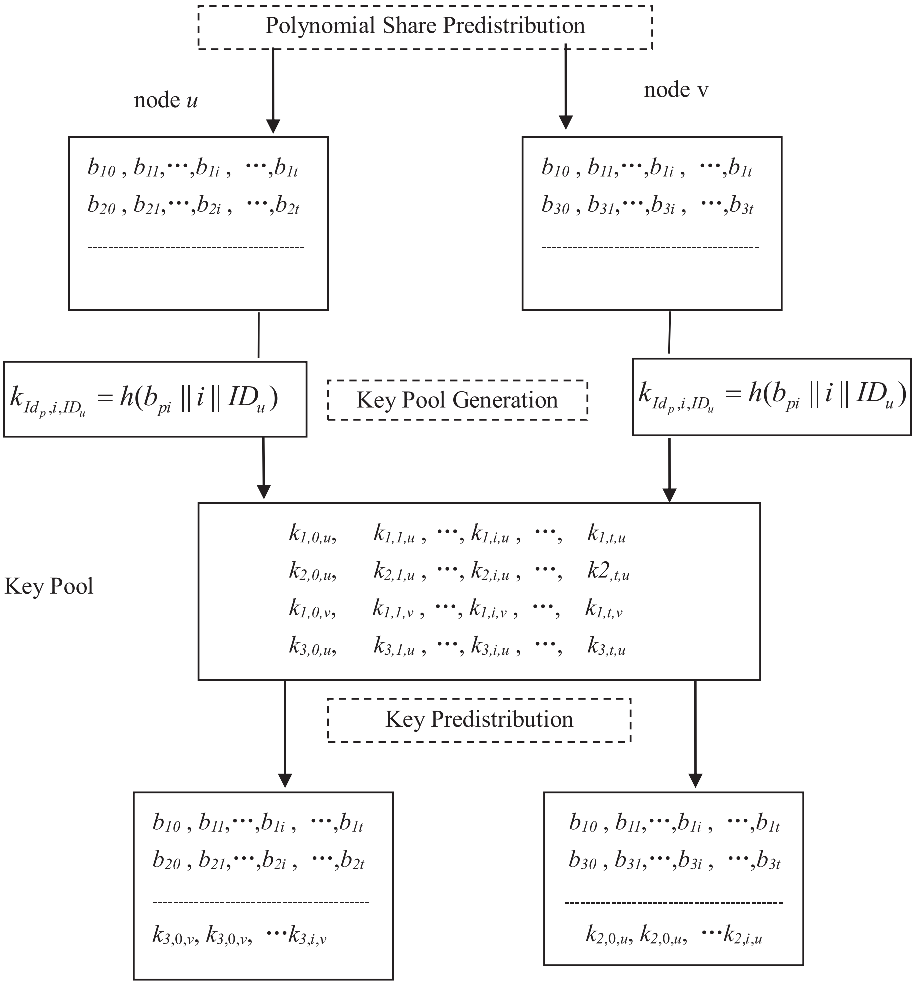

Polynomial Share Predistribution
node u
node v
$b_{10}, b_{11},\cdots,b_{1i}, \cdots,b_{1t}$
$b_{20}, b_{21},\cdots,b_{2i}, \cdots,b_{2t}$
$b_{10}, b_{11},\cdots,b_{1i}, \cdots,b_{1t}$
$b_{30}, b_{31},\cdots,b_{3i}, \cdots,b_{3t}$
$k_{Id_p,i,ID_u} = h(b_{pi} \| i \| ID_u)$
Key Pool Generation
$k_{Id_p,i,ID_u} = h(b_{pi} \| i \| ID_u)$
Key Pool
$k_{1,0,u}, k_{1,1,u}, \cdots, k_{1,i,u}, \cdots, k_{1,t,u}$
$k_{2,0,u}, k_{2,1,u}, \cdots, k_{2,i,u}, \cdots, k2_{,t,u}$
$k_{1,0,v}, k_{1,1,v}, \cdots, k_{1,i,v}, \cdots, k_{1,t,v}$
$k_{3,0,u}, k_{3,1,u}, \cdots, k_{3,i,u}, \cdots, k_{3,t,u}$
Key Predistribution
$b_{10}, b_{11},\cdots,b_{1i}, \cdots,b_{1t}$
$b_{20}, b_{21},\cdots,b_{2i}, \cdots,b_{2t}$
$k_{3,0,v}, k_{3,0,v}, \cdots k_{3,i,v}$
$b_{10}, b_{11},\cdots,b_{1i}, \cdots,b_{1t}$
$b_{30}, b_{31},\cdots,b_{3i}, \cdots,b_{3t}$
$k_{2,0,u}, k_{2,0,u}, \cdots k_{2,i,u}$

Fig. 1. A sample of the initial phase.

to encrypt the identification of the key information generated by the communication key, such as the identification of sensor nodes, the identification of polynomials and the identification of the keys. When the node receives the encrypted message, it uses the communication key to decrypt the encrypted text and compare the received information with that previously received. If these identifications are equal, the received identification of the key generation information is verified.

Additionally, in the three cases, the method will first check the validity of Case 1. If Case 1 is true, it will not check Cases 2 or 3. If Case 2 is true, it will not check Case 3. According to the communication-establishing mechanism, the communication keys computed under Case 1 are *B*-communication-keys and the communication keys computed under Cases 2 and 3 are *R*-communication-keys.

If two nodes can't establish the communication key according the above methods, they may resort to the path key establishment defined in the RKP method to compute their shared key.

5. Performance analysis and comparison

In this section, we evaluate the performance of the proposed scheme using the following properties: storage, the communication and computational overhead of every sensor node, network connectivity, and the resilience against sensor node capture. In the following subsections, we use Matlab to plot the results.

5.1. Overhead

To analyse the overheads of the proposed and related schemes, the following parameters are considered.

- l_k is the length of a key. Here, we assume that each generated key takes the same storage as the coefficient of a t-degree polynomial.
- l_{ID} is the length of a node or key identifier. Here, we assume that the node and key identifiers have the same length.
- d is the average number of the neighbouring nodes for each node.
- r is the number of keys selected for each node in the RKP and q-composite schemes.
- n is the number of polynomials selected for each node in the DDHV scheme.

5.1.1. Storage overhead

According to the initialization phase of the proposed scheme, the preloaded materials in a sensor include s keys, the identification of the keys, the shares of g polynomials, the identification of the polynomials, and the identification of the sensor nodes. Thus, the overall storage overhead of each sensor is $(2s+g+1)$ $l_{ID}+(s+g(t+1))\ l_k$.

In the RKP and q-composite schemes (for $q=2$ and 3), each node only needs to store r keys and their identifiers. Then, the storage overhead of each sensor in these schemes is rl_k+rl_{ID}. In the DDHV scheme, each node needs to store the shares of

Table 2
Comparisons of overheads.

Scheme	Storage overhead	Communication overhead	Computational overhead
RKP scheme	$rl_k + rl_{ID}$	rl_{ID}	dr log r CPs
q-composite scheme	$rl_k + rl_{ID}$	rl_{ID}	drlog r CPs + d HOs
DDHV scheme]	$(n(t+1))l_k + (n+1)l_{ID}$	$(n+1)l_{ID}$	dnlog n CPs + d PEs
Our scheme	$(s+g(t+1))\ l_k + (2s+g+1)\ l_{ID}$	$(2s+g+1)\ l_{ID}$	dg log g CPs + d PEs or $d(g$ log $g+s$ log $s)$ CPs or $d(g$ log $g+s$ log $s+$sg/2)CP + dHOs

Note: CP: comparison, HO: hash operation, PE: t-degree polynomial evaluation.

Table 3
Comparisons of overheads in a given key data storage.

Scheme	Storage overhead	Communication overhead	Computational overhead
RKP scheme	$200l_k + 200l_{ID}$	$200l_{ID}$	200d log200 CPs
q-composite scheme	$200l_k + 200l_{ID}$	$200l_{ID}$	200d log200 CPs + d HOs
DDHV scheme	$200l_k + 3l_{ID}$ (t = 99)	$3l_{ID}$	3$d$log 3 CPs + d PEs
Our scheme	$200l_k + 159l_{ID}$	$159l_{ID}$	2d log2g CPs + d PEs or $d(2$ log $2+78$log $78)$ CPs or $d(2$ log $2+78$ log $78+78)$ CP + dHOs

Note: C: comparison, HO: hash operation, PE: t-degree polynomial evaluation.

m polynomials and the identification of the polynomials. The storage overhead of each sensor node in the DDHV scheme is $(m+1)l_{ID} + (m(t+1))l_k$.

5.1.2. Communication and computational overhead

The path key establishment is a complicated procedure. It requires more communication and computational overhead for the establishment of path keys between neighbouring nodes. In the initialization phase, there is no communication needed and all computations can be executed by the KDS. In this paper, we concentrate on only the direct key establishment phases of different schemes when analysing communication and computational overheads.

In terms of communication overhead, during the direct communication establishment phase, each sensor needs to disclose its ID, the identification of s pre-loaded keys and g pre-loaded polynomials to its neighbour nodes, which is $(3s+g+1)l_{ID}$ bits.

In terms of computational overhead, the computations include the number of comparisons when identifying the common key(s) or polynomial(s), the number of polynomial evaluations, and the number of hash operations. We assume that the IDs of keys or polynomials are stored in ascending order in each node and a binary search is performed to locate the ID of the common key or polynomial. The computational overheads are shown in Table 2.

In these schemes, data storage in memory includes key data (keys or polynomial shares) and ID data (the IDs of keys, polynomial shares or the sensor node). In the key distribution schemes, a lower memory requirement implies a better scheme.

In Table 3, we use the same amount of memory for storing key data per node in order to conduct a fair comparison. Here, we suppose that the amount of memory needed for storing keys or polynomials shares is 200 l_k. In the proposed scheme, $(s+g(t+1))$ $l_k = 200$ l_k. If $t = 60$ and $g = 2$, then $s = 78$. In the DDHV scheme, $n(t+1))l_k = 200$ l_k.f $t = 99$, then $n = 3$.

From Table 3, we observe that the storage, communication and computational overheads required by our scheme are comparable to those of the RKP, q-composite and DDHV schemes.

5.2. Local connectivity

Local connectivity is the probability of establishing communications between the nodes located within a direct communication range. It is an important metric to evaluate a key predistribution scheme. To achieve a desired global connectivity, the local connectivity must be higher than a certain value. This probability will be denoted by P_L. In general, we want P_L to be large.

Suppose $B(u,v)$ is the event and the communication key between nodes u and v is a B-communication key. $R(u,v)$ is the event and the communication key between node u and v is a R-communication key. Then, we have

$$P_L = P(B(u,v) + R(u,v)). \tag{1}$$

Let $E_1(u,\ v)$ be the event in which sensor nodes u and v have the same polynomial. $E_2(u,\ v)$ is the event in which sensor nodes u and v have the same predistributed keys. $E_3(u,\ v)$ is the event in which the identification Id_p of the predistributed polynomial in either node u or v is the same as the tuple component Id_p in the identification of predistributed keys in another node. According to the previous section, we have

$$B(u,v) = E_1(u,v) \tag{2}$$

$$\begin{aligned} R(u,v) &= \overline{E_1(u,v)}\Big(E_2(u,v) + \overline{E_2(u,v)} \cdot E_3(u,v)\Big) \\ &= \overline{E_1(u,v)} \cdot E_2(u,v) + \overline{E_1(u,v)} \cdot \overline{E_2(u,v)} \cdot E_3(u,v). \end{aligned} \tag{3}$$

As the events $B(u,v)$ and $R(u,v)$ are two mutually exclusive events, we get the local connective of node u and v as follows:

$$\begin{aligned} P_L &= P(B(u,v) + R(u,v)) \\ &= P\Big(E_1(u,v) + \overline{E_1(u,v)} \cdot E_2(u,v) + \overline{E_1(u,v)} \cdot \overline{E_2(u,v)} \cdot E_3(u,v)\Big) \\ &= P(E_1(u,v)) + P\Big(\overline{E_1(u,v)} \cdot E_2(u,v)\Big) \\ &\quad + P\Big(\overline{E_1(u,v)} \cdot \overline{E_2(u,v)} \cdot E_3(u,v)\Big) \\ &= P(E_1(u,v)) + P\Big(\overline{E_1(u,v)} \cdot E_2(u,v)\Big) \\ &\quad + P\Big(\overline{E_1(u,v)} \cdot \overline{E_2(u,v)} \cdot E_3(u,v)\Big) \\ &= 1 - P\Big(\overline{E_1(u,v)}\Big) + P\Big(\overline{E_1(u,v)}\Big)P\Big(E_2(u,v)|\overline{E_1(u,v)}\Big) \\ &\quad + P\Big(\overline{E_1(u,v)} \cdot \overline{E_2(u,v)} \cdot E_3(u,v)\Big) \\ &= 1{-}P\Big(\overline{E_1(u,v)}\Big)\Big(\overline{E_2(u,v)}|\overline{E_1(u,v)}\Big) \\ &\quad + P\Big(\overline{E_1(u,v)}\Big)P\Big(\overline{E_2(u,v)} \cdot E_3(u,v)|\overline{E_1(u,v)}\Big) \\ &= 1{-}P\Big(\overline{E_1(u,v)}\Big)P\Big(\overline{E_2(u,v)} \cdot \overline{E_3(u,v)}|\overline{E_1(u,v)}\Big). \end{aligned} \tag{4}$$

As there are m polynomials and each node picks g polynomials, as computed in [9],

$$P(\overline{E_1(u,v)}) = \frac{\binom{m}{g}\binom{m-g}{g}}{\binom{m}{g}\binom{m}{g}} = \frac{\binom{m-g}{g}}{\binom{m}{g}}. \tag{5}$$

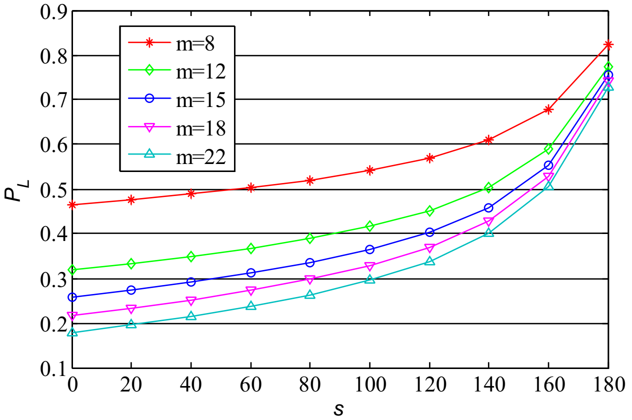


Fig. 2. Local connectivity versus the number of keys in each sensor key s in a node and the polynomial pool size m, given $N=2000$, $g=2$, and $w=200$.

As there are N nodes in the WSNs and each node will generate $g(t+1)$ keys, there are $Ng(t+1)$ keys in the key pool. Then, we have,

$$P\left(\overline{E_2(u,v)}\cdot\overline{E_3(u,v)}|\overline{E_1(u,v)}\right)$$
$$=\frac{\binom{Ng(t+1)-2g(t+1)}{s}\binom{Ng(t+1)-2g(t+1)-s}{s}}{\binom{Ng(t+1)-g(t+1)}{s}\binom{Ng(t+1)-g(t+1)}{s}}$$
$$=\frac{\binom{g(t+1)(N-2)}{s}\binom{g(t+1)(N-2)-s}{s}}{\binom{g(t+1)(N-1))}{s}\binom{g(t+1)(N-1)}{s}}. \tag{6}$$

From (1) to (3), we have

$$P_L = 1-P\left(\overline{E_1(u,v)}\right)P\left(\overline{E_2(u,v)}|\left(\overline{E_1(u,v)}\right)\right)$$
$$=1-\frac{\binom{m-g}{g}}{\binom{m}{g}}\frac{\binom{g(t+1)(N-2)}{s}\binom{g(t+1)(N-2)-s}{s}}{\binom{g(t+1)(N-1)}{s}\binom{g(t+1)(N-1)}{s}}. \tag{7}$$

As each node is preloaded with the polynomial shares of g polynomials and s keys, we have

$$g(t+1)+s=w. \tag{8}$$

Then,

$$P_L = 1-\frac{\binom{m-g}{g}}{\binom{m}{g}}\frac{\binom{g(t+1)(N-2)}{w-g(t+1)}\binom{g(t+1)(N-2)-w-g(t+1)}{w-g(t+1)}}{\binom{g(t+1)(N-1)}{w-g(t+1)}\binom{g(t+1)(N-1)}{w-g(t+1)}}$$
$$=1-\frac{\binom{m-g}{g}}{\binom{m}{g}}\frac{\binom{(w-s)(N-2)}{s}\binom{(w-s)(N-2)-s}{s}}{\binom{(w-s)(N-1)}{s}\binom{(w-s)(N-1)}{s}}. \tag{9}$$

Fig. 2 indicates the local connectivity according to s and m in the given value of g. It is shown that, under the same m, the local network connectivity will increase with an increasing number of preloaded keys. This is because, in a given memory, when the number of preloaded keys increases, the preloaded polynomials will decrease and the generated keys in the key pool will also decrease. Fig. 3 illustrates the relationship between local network connectivity and the combinations of N, g, and t. From Figs. 2 to 3, we know that we can get the needed network size and local connectivity by selecting the proper parameters.

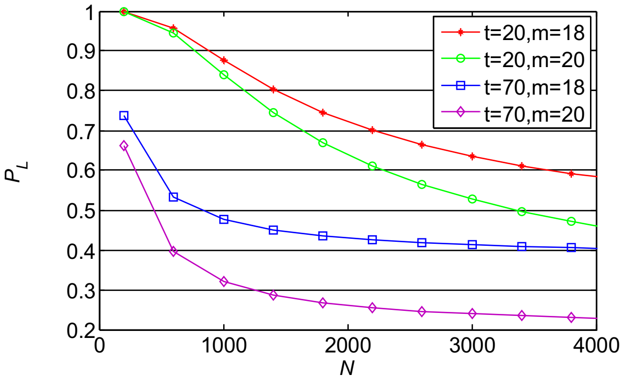


Fig. 3. The local connectivity versus the number of nodes in a WSN and the combination of m and t, given $g=2$ and $w=200$.

5.3. Security analysis

Because sensor nodes are not assumed to be tamper resistant, an adversary may physically capture sensor nodes and compromise the stored secret information. In the context of key distribution, the adversary can infer the communication key shared between the uncompromised sensor nodes. Thus, the resistance against sensor node compromising attacks should be considered when designing the predistribution schemes.

In the following section, we evaluate how the proposed scheme improves network security in terms of resilience against node capture. Then, we compare our scheme to some existing schemes.

5.3.1. Resilience against node capture

We evaluate the resilience of networks by computing the probability that a random link is broken by compromising a set of x random nodes not in the link for suitable values of x. We denote this probability as Pc. In general, we want Pc to be small.

Suppose K is the communication key used by two non-compromised nodes u and v. EB is the joint event that the key K is a B-communication-key KB and KB has been compromised., ER is the event that the communication key is a R-communication-key KR and KR has been compromised. Here, the notation $K\in KB$ represents that "Key K was a B-communication-key" and $K\in KR$ represents that "Key K was a R-communication –key."

Assume that C_x is the event that x nodes have been compromised. Then, when x nodes have been compromised, the probability that key K has been compromised is:

$$P_c = P(K_{compromised}|C_x) = P(EB\cup ER|C_x). \tag{10}$$

From the definition of events EB and ER, we note that these two events are mutually exclusive. Then,

$$P_c = P(EB|C_x) + P(ER|C_x). \tag{11}$$

Since the event $K\in KB$ is dependent on event C_x (KB is compromised) and the event $K\in KR$ is dependent on event C_x (KR is compromised), we have

$$
\begin{aligned}
P(K_{compromised}|C_x) &= P(EB|C_x) + P(ER|C_x)\\
&= P(((K \in KB) \cap (KB\ is\ compromised))|C_x)\\
&\quad + P(((K \in KR) \cap (Kr\ is\ compromised))|C_x)\\
&= \frac{P((K \in KB) \cap (KB\ is\ compromised) \cap C_x)}{P(C_x)}\\
&\quad + \frac{P((K \in KR) \cap (KR\ is\ compromised) \cap C_x)}{P(C_x)}\\
&= P(K \in KB)P(KB\ is\ compromised|C_x)\\
&\quad + P(K \in KR)P(KR\ is\ compromised|C_x). \qquad (12)
\end{aligned}
$$

From the Section 4.2 we get

$$
P(K \in KB) = \frac{P(B(u,v))}{P_L} = \frac{P(E_1(u,v))}{P_L} = \frac{1-P\left(\overline{E_1(u,v)}\right)}{P_L} \qquad (13)
$$

$$
\begin{aligned}
P(K \in KR) &= \frac{P(R(u,v))}{P_L} = \frac{P_L - P(B(u,v))}{P_L}\\
&= \frac{P_L - 1 + P\left(\overline{E_1(u,v)}\right)}{P_L}. \qquad (14)
\end{aligned}
$$

As the preloaded keys in nodes are one-way functional values of the polynomial shares, the adversary can't infer polynomial shares from the preloaded keys in the compromised nodes. Similar to the analysis in [7,9], we have

$$
P(KB\ is\ compromised|C_x) = 1 - \sum_{i=0}^{t}\binom{x}{i}\left(\frac{g}{m}\right)^{i}\left(1-\frac{g}{m}\right)^{x-i}. \qquad (15)
$$

$$
P(KR\ is\ compromised|C_x) = 1 - \left(1 - \frac{g(t+1)+s}{Ng(t+1)}\right)^{x}. \qquad (17)
$$

From formulas (11)–(17), the probability that the communication key K between two noncompromised sensors is compromised is:

$$
\begin{aligned}
Pc &= \left(1 - \sum_{i=0}^{t}\binom{x}{i}\left(\frac{g}{m}\right)^{i}\left(1-\frac{g}{m}\right)^{x-i}\right)\left(\frac{1-P\left(\overline{E_1(u,v)}\right)}{P_L}\right)\\
&\quad + \left(1-\left(1-\frac{(t+1)+s}{Ngt}\right)^{x}\right)\left(\frac{P_L-1+P\left(\overline{E_1(u,v)}\right)}{P_L}\right)\\
&= \left(\left(1-\frac{(t+1)+s}{Ngt}\right)^{x} - \sum_{i=0}^{t}\binom{x}{i}\left(\frac{g}{m}\right)^{i}\left(1-\frac{g}{m}\right)^{x-i}\right)\\
&\quad \times \left(\frac{1-P\left(\overline{E_1(u,v)}\right)}{P_L}\right) + \left(1-\left(1-\frac{(t+1)+s}{Ngt}\right)^{x}\right). \qquad (18)
\end{aligned}
$$

Fig. 4 shows the relationship between the fraction of compromised links for noncompromised nodes and the number of compromised nodes. In this figure, every sensor node is capable of holding 200 cryptographic keys in its memory and there are 2000 sensor nodes in the network. From the figure, we can see that if a larger t leads to a resilience against node capture closer to the DDHV scheme in [15]. A smaller t leads to a resilience against node capture closer to the RKP scheme in [13]. From formula (5), we see that a larger t leads to a smaller s. A smaller s indicates that there are more polynomial shares preloaded in the nodes and the properties of the proposed scheme will be closer to the DDHV scheme.

5.3.2. Comparison with previous schemes

In this subsection, we compare the resistance against node capture attacks of the proposed scheme with that of related works, including the RKP scheme [13], the q-composite scheme (for $q=2$ and 3) [14], and the DDHV scheme [15].

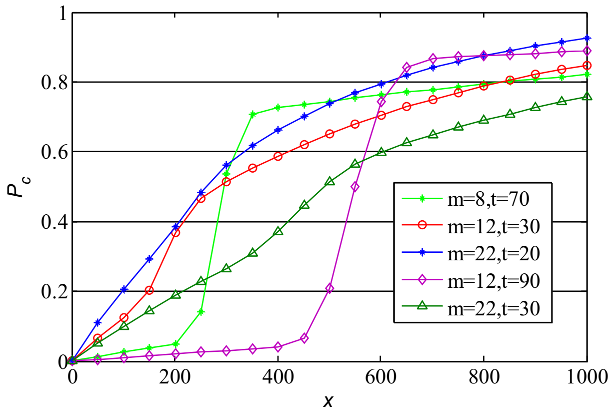
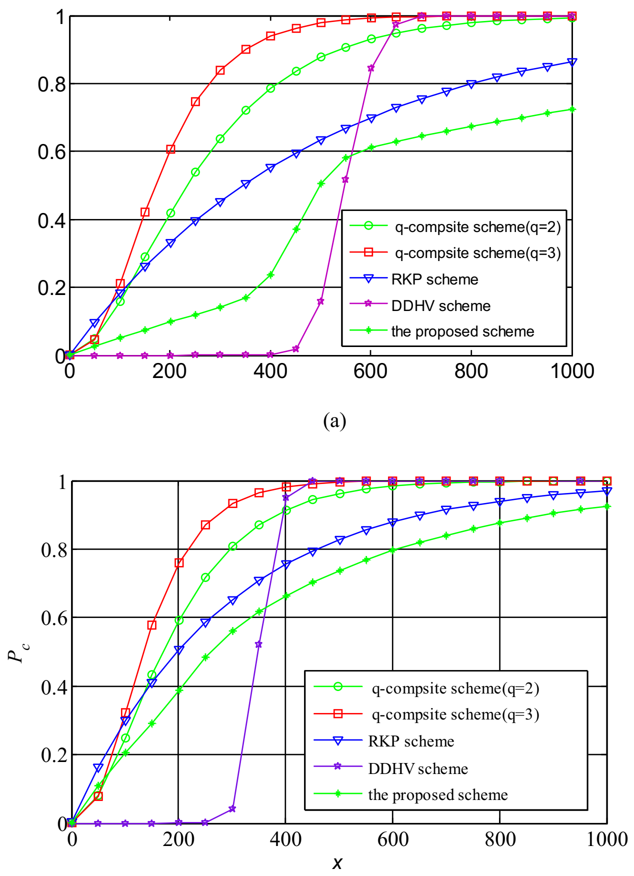


Fig. 4. Fraction of compromised links between non-compromised nodes with different connectivity after an adversary has compromised x random nodes.



(a)

(b)

Fig. 5. Comparison between schemes with different local connectivity. Assume each node has available storage for 200 keys. (a) $Pc=0.34$. (b) $Pc=0.5$.

To have fair comparisons, we use the same amount of the storage per node in all schemes. In the following simulations, we assume that each sensor node can store 200 keys in its memory.

Fig. 5 illustrates the comparisons of different schemes under the same local connectivity P_L with suitable parameter values for the different schemes. The results show that our scheme offers better security compared to the other three schemes. It is very clear from the figures that our scheme has better resilience against node capture attacks than the *RKP* scheme, the q-composite $(q=2)$ scheme, and the q-composite $(q=3)$ scheme, even if the number of captured nodes is small. For the DDHV scheme, as long as the number of captured nodes remains below a threshold value, it performs well. However, once the number of captured nodes is greater than the threshold value, the attacker will be able to infer all the communication keys between noncompromised nodes. As shown in Fig. 5(a), if attackers capture 600 sensor nodes, 70.0% of the communication keys between noncompromised nodes will be compromised. If attackers capture 700 nodes, all communication keys between noncompromised nodes will be compromised. In the proposed scheme, only 64.5% of communication keys between noncompromised nodes will be compromised, even if the attackers capture 700 nodes.

5.3.3. Security proof using BAN-Logic

As the nodes exchange the key generation information through an open channel, the attackers may forge the key generation information to cheat the nodes. Now we use Burrows–Abadi–Needham Logic [32], generally called BNN logic, to prove that our scheme can prevent attackers from sending falsely identified key generation information.

The notations used in rest of the paper are listed in Table 3.

The BAN logic is a well-known formal model used to analyse the security of authentication and key distribution protocols [33,34]. Some BAN statements that are helpful for analysing the security of the proposed method are given below.

$P| \equiv X$: P believes X, or P would be entitled to believe X. In particular, P can take X as true.

$P \lhd X$: P sees X. P has received some message X and is capable of reading and repeating it.

$P|\sim X$: P once said X. P at some time sends a message including the statement X. It is known whether this is a repeated message, although it is known that P believed X when he sent it.

$\#(X)$: The message X is fresh.

(X, Y): The formula X or Y are one part of the formulae (X, Y).

$< X >_K$: The formula X is encrypted under the key K.

$P \overset{K}{\longleftrightarrow} Q$: Principals P and Q communicate via shared key K.

Some main logical postulates of the BAN logic are listed below and will be used in our analysis.

The message-meaning rule: $\frac{P|\equiv P \overset{K}{\longleftrightarrow} Q, P \lhd <X>_K}{P|\equiv Q|\sim X}$.

If principal P believes that secret K is shared with Q and sees$<X>K$, the P believes that Q once said X.

The session keys rule: $\frac{P|\equiv \#(X), P|\equiv Q|\equiv X}{P|\equiv P \overset{K}{\longleftrightarrow} Q}$.

If principal P believes that the session key is fresh and principals P and Q believe X (which are the necessary parameters of the session key), then principal P shares the session key K with Q.

To prove that the proposed protocol is secure, the proposed protocol must satisfy the following goals that are based on the BAN logic and given below.

Goal 1: $S_1|\equiv S_2|\sim(ID_2, IP_i)$, **Goal 2:** $S_2|\equiv S_1|\sim(ID_1, IP_i)$
Goal 3: $S_3|\equiv S_4|\sim(IP_i, j, ID_k)$, **Goal 4:** $S_4|\equiv S_3|\sim(IP_i, j, ID_k)$
Goal 5: $S_1|\equiv S_3|\sim(IP_i, j, ID_k)$, **Goal 6:** $S_3|\equiv S_1|\sim IP_i$

First, the following assumptions are made about the initial status of our scheme.

A1: $S_1| \equiv S_2| \equiv (ID_2, IP_i)$, A2: $S_2| \equiv S_1| \equiv (ID_1, IP_i)$
A3: $S_3| \equiv S_4| \equiv (IP_i, j, ID_k)$, A4: $S_4| \equiv S_3| \equiv (IP_i, j, ID_k)$
A5: $S_1| \equiv S_3| \equiv (IP_i, j, ID_k)$, A6: $S_3| \equiv S_1| \equiv IP_i$

Second, the proposed protocol is transformed into an idealized form.

M1: $S_2 \to S_1$: (ID_2, IP_i), M2: $S_1 \to S_2$: (ID_1, IP_i)
M3: $S_2 \to S_1$: $< ID_2, IP_i >_{SK21}$, M4: $S_1 \to S_2$: $< ID_1, IP_i >_{SK12}$
M5: $S_4 \to S_3$: (IP_i, j, ID_k), M6: $S_3 \to S_4$: (IP_i, j, ID_k)
M7: $S_4 \to S_3$: $< IP_i, j, ID_k >_{SK43}$, M8: $S_3 \to S_4$: $< IP_i, j, ID_k >_{SK34}$
M9: $S_3 \to S_1$: (IP_i, j, ID_k), M10: $S_1 \to S_1$: IP_i
M11: $S_3 \to S_1$: $< IP_i, j, ID_k >_{SK13}$, M12: $S_1 \to S_3$: $< IP_i >_{SK13}$

Third, the idealized form of the proposed scheme is analysed using the BAN logic. The main steps are described as follows.

From M1, we have

D1: $S_1| \equiv \#(ID_2, IP_i)$.

From A1, D1 and the session keys rule, we obtain

D2: $S_1| \equiv S_1 \overset{SK_{12}}{\longleftrightarrow} S_2$.

From M3, we have

D3: $S_1 \lhd < ID_2, IP_i >_{SK_{21}}$.

From D2, D3 and the message-meaning rule, we get

D4: $S_1|\equiv S_2|\sim(ID_2, IP_i)$. (**Goal 1**)

From M2, we have

D5: $S_2| \equiv \#(ID_1, IP_i)$.

From A2, D5 and the session keys rule, we obtain

D6: $S_2| \equiv S_2 \overset{SK_{12}}{\longleftrightarrow} S_1$.

From M3, we have

D7: $S_1 \lhd < ID_2, IP_i >_{SK_{12}}$.

From D6, D7, and the message-meaning rule, we get

D8: $S_2|\equiv S_1|\sim(ID_1, IP_i)$. (**Goal 2**)

From M5, we have

D9: $S_3| \equiv \#(IP_i, j, ID_k)$.

From A3, D9 and the session keys rule, we obtain

D10: $S_3| \equiv S_4 \overset{SK_{34}}{\longleftrightarrow} S_3$.

From M7, we have

D11: $S_3 \lhd < IP_i, j, ID_k >_{SK_{34}}$.

From D10, D11, and the message-meaning rule, we get

D12: $S_3|\equiv S_4|\sim(IP_i, j, ID_k)$. (**Goal 3**)

From M7, we have

D13: $S_4| \equiv \#(IP_i, j, ID_k)$.

From A4, D13 and the session keys rule, we obtain

D14: $S_4| \equiv S_3 \overset{SK_{12}}{\longleftrightarrow} S_4$.

From M8, we have

D15: $S_4 \lhd < IP_i, j, ID_k >_{SK_{34}}$.

From D14, D15, and the message-meaning rule, we get

D16: $S_4| \equiv S_3| \sim (IP_i,\ j,\ ID_k)$. (**Goal 4**)

From M9, we have

D17: $S_1| \equiv \#(IP_i,\ j, ID_k)$.

From A5, D17 and the session keys rule, we obtain

D18: $S_1| \equiv S_3 \overset{SK_{13}}{\longleftrightarrow} S_1$.

From M11, we have

D19: $S_1 \triangleleft <IP_i,\ j, ID_k>_{SK_{13}}$.

From D18, D19, and the message-meaning rule, we get

D20: $S_1| \equiv S_3| \sim (IP_i,\ j,\ ID_k)$. (**Goal 5**)

From M10, we have

D21: $S_1| \equiv \#(IP_i)$.

From A6, D21 and the session keys rule, we obtain

D22: $S_3| \equiv S_3 \overset{SK_{13}}{\longleftrightarrow} S_1$.

From M8, we have

D23: $S_3 \triangleleft <IP_i>_{SK_{13}}$.

From D22, D23 and the message-meaning rule, we get

D24: $S_1| \equiv S_2| \sim IP_i$. (**Goal 6**)

6. Conclusion

This paper presents a new key predistribution approach for WSNs. This scheme was based on the random key predistribution and polynomial-base key predistribution scheme. In the proposed scheme, the predistributed keys in compromised sensor nodes will not disclose any information regarding the polynomial shares in the uncompromised sensor nodes. The effectiveness of the proposed scheme has been demonstrated through analyses and simulations. The simulation results show that the proposed scheme provides better resilience against node capture attacks compared to previous schemes.

Two directions are worth future research. First, it might be worthwhile to extend the combination of the DDHV scheme analysis to key predistribution schemes other than the RKP scheme. A good candidate would be the q-composite scheme introduced in [14], which is a direct extension of the RKP scheme. Second, we would extend this method to cluster-based WSNs.

References

[1] K. Romer, F. Mattern, The design space of wireless sensor networks, *IEEE Wireless Commun.* 11 (6) (2004) 54–61.

[2] R.P. Kurbah, B. Sharma, Survey on issues in wireless sensor networks: attacks and countermeasures, Int. J. Comput. Sci. Inf. Secur. 14 (4) (2016) 262.

[3] D. Puccinelli, M. Haenggi, Wireless sensor networks: applications and challenges of ubiquitous sensing, *IEEE Circuits Syst. Mag.* 5 (3) (2005) 19–31.

[4] D.S. Kridi, C.G.N. de Carvalho, D.G. Gomes, Application of wireless sensor networks for beehive monitoring and in-hive thermal patterns detection, *Comput. Electron. Agric.* 127 (2016) 221–235.

[5] D. He, C. Chen, S. Chan, J. Bu, L.T. Yang, Security analysis and improvement of a secure and distributed reprogramming protocol for wireless sensor networks, *IEEE Trans. Ind. Electron.* 60 (11) (2013) 5348–5354.

[6] H. Song, G.A. Fink, G.L. Rosner, S. Jeschke, Security and Privacy in Cyber-Physical Systems: Foundations, Principles and Applications, 2017.

[7] R.V. Saraswathi, L.P. Sree, K. Anuradha, Key management schemes in wireless sensor networks: a survey, *Wireless Commun.* 8 (5) (2016) 185–190.

[8] M. Ge, K.K.R. Choo, H. Wu, Y. Yu, Survey on key revocation mechanisms in wireless sensor networks, *J. Netw. Comput. Appl.* 63 (2016) 24–38.

[9] R. Ezhilarasie, A. Umamakeswari, T. Renugadevi, Key management schemes in wireless sensor networks: a survey, *Int. J. Adv. Intell. Paradigms* 7 (3-4) (2015) 222–239.

[10] L. Ya-Nan, W. Jian, D. He, S. Li-Jun, Intra-cluster key sharing in hierarchical sensor networks, *IET Wireless Sensor Syst.* 3 (3) (2013) 172–182.

[11] Q. Mamun, R. Islam, M. Kaosar, Secured communication key establishment for cluster-based wireless sensor networks, *Int. J. Wireless Netw. Broadband Technol.* 4 (1) (2015) 29–44.

[12] A. Mehmood, M.M. Umar, H. Song, ICMDS: secure inter-cluster multiple-key distribution scheme for wireless sensor networks, *Ad Hoc Netw.* 55 (2017) 97–106.

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

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

[15] D. Liu, P. Ning, R. Li, Establishing pairwise keys in distributed sensor networks, *ACM Tran. Inf. Syst. Secur.* 8 (1) (2005) 41–77.

[16] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, Perfectly-secure key distribution for dynamic conferences, in: Annual International Cryptology Conference, August, Springer Berlin Heidelberg, 1992, pp. 471–486.

[17] W. Du, J. Deng, Y.S. Han, P.K. Varshney, J. Katz, A. Khalili, A pairwise key predistribution scheme for wireless sensor networks, *ACM Trans. Inf. Syst. Secur.* 8 (2) (2005) 228–258.

[18] A. Rasheed, R. Mahapatra, Key predistribution schemes for establishing pairwise keys with a mobile sink in sensor networks, *IEEE Trans. Parallel Distrib. Syst.* 22 (1) (2011) 176–184.

[19] H. Dai, H. Xu, Key predistribution approach in wireless sensor networks using LU matrix, *IEEE Sens. J.* 10 (8) (2010) 1399–1409.

[20] E. Baburaj, Polynomial and multivariate mapping-based triple-key approach for secure key distribution in wireless sensor networks, Computers & Electrical Engineering (2016).

[21] D Liu, P Ning, Improving key predistribution with deployment knowledge in static sensor networks, *ACM Trans. Sensor Netw.* 1 (2) (2005) 204–239.

[22] W. Du, J. Deng, Y.S. Han, P.K. Varshney, A key predistribution scheme for sensor networks using deployment knowledge, *IEEE Trans. Dependable Secure Comput.* 3 (1) (2006) 62–77.

[23] J. Choi, Y. Kim, J. Kim, T. Kwon, A study of location-based key management using a grid for wireless sensor networks, *J. Korea Inst. Inf. Secur. Cryptol.* 25 (4) (2015) 759–766.

[24] J. Lee, T. Kwon, GenDep: location-aware key management for general deployment of wireless sensor networks, *Int. J. Distrib. Sens. Netw.* (2014).

[25] T. Kwon, J. Lee, J. Song, Location-based pairwise key predistribution for wireless sensor networks, *IEEE Trans. Wireless Commun.* 8 (11) (2009).

[26] S.A. Çamtepe, B. Yener, Combinatorial design of key distribution mechanisms for wireless sensor networks, *IEEE/ACM Trans. Netw.* 15 (2) (2007) 346–358.

[27] C. Castelluccia, A. Spognardi, Rok: a robust key pre-distribution protocol for multi-phase wireless sensor networks, in: Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on, September, IEEE, 2007, pp. 351–360.

[28] A.K. Das, A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks, *Int. J. Inf. Secur.* 11 (3) (2012) 189–211.

[29] O. Catakoglu, A. Levi, Uneven key pre-distribution scheme for multi-phase wireless sensor networks, in: Information Sciences and Systems 2013, Springer International Publishing, 2013, pp. 359–368.

[30] B. Zhou, J. Wang, S. Li, W. Wang, A new key predistribution scheme for multi-phase sensor networks using a new deployment model, J. Sensors 2014 (2014).

[31] M.L. Messai, H. Seba, A survey of key management schemes in multi-phase wireless sensor networks, Comput. Netw. 105 (2016) 60–74.

[32] M Burrows, M Abadi, M Needham R, A logic of authentication, in: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, The Royal Society, 1989, pp. 233–271.

[33] R Amin, G.P. Biswas, Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment, Wireless Pers. Commun. 84 (1) (2015) 439–462.

[34] R. Ali, A.K. Pal, S. Kumari, et al., A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring, Fut. Gen. Comput. Syst. (2017).

Jianmin Zhang received the Ph.D. degree from Huazhong Science and Technology University, China in 2007. Now he is the associate professor of College of Computer, Henan Institute of Engineering. His current research interests include wireless sensor networks, network security.

Hua Li received the M.S. degree from Zhengzhou University, China. Now he is the lecture of College of Computer, Henan Institute of Engineering and his research interests include wireless sensor networks, network security.

Jian Li received the M.S. degree from Henan University, China. He is currently a professor and vice-president of College of Computer at Henan Institute of Engineering .His current research interests include algorithm design and analysis, wireless sensor networks, software engineering, grid computing.