Bag and Roy *EURASIP Journal on Wireless Communications and Networking* 2013, **2013**:145
http://jwcn.eurasipjournals.com/content/2013/1/145

EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

A new key predistribution scheme for general and grid-group deployment of wireless sensor networks

Samiran Bag* and Bimal Roy

Abstract

Key predistribution for wireless sensor networks has been a challenging field of research because stringent resource constraints make the key predistribution schemes difficult to implement. Despite this, key predistribution scheme is regarded as the best option for key management in wireless sensor networks. Here, the authors have proposed a new key predistribution scheme. This scheme exhibits better performance than existing schemes of its kind. Moreover, our scheme ensures constant time of key establishment between two nodes. We provide some bounds on the resiliency of this scheme.

Next, we use this new key predistribution scheme in a grid-group deployment of sensor nodes. The entire deployment zone is broken into square regions. The sensor nodes falling within a single square region can communicate directly. Sensor nodes belonging to different square regions can communicate by means of special nodes deployed in each of the square region. We measure the resiliency in terms of fraction of links disconnected as well as fraction of nodes and regions disconnected. We show that our key predistribution scheme when applied to grid-group deployment performs better than standard models in existence.

1 Introduction

Key predistribution in wireless sensor networks has attracted attention of researchers for a decade. Key predistribution schemes are classified into two groups viz. probabilistic key predistribution and deterministic key predistribution. In probabilistic key predistribution scheme, as the name implies, the keys are randomly drawn from a large pool of keys and are placed into the individual sensor nodes. This scheme does not ensure full connectivity between nodes. However, due to this scheme's randomness, it does ensure resiliency against selective node capture attack. Some probabilistic schemes can be found in [1-3]. The main disadvantage of probabilistic key predistribution schemes are that they do not ensure full connectivity between each and every pair of nodes. On the other hand, in deterministic key predistribution scheme, a deterministic method is employed to load the keys into the sensor nodes. This scheme may or may not offer full connectivity between every pair of nodes of the Wireless Sensor Network (WSN). Several deterministic key predistribution schemes have been proposed by researchers. Blom [4] proposed a scheme for key for pairwise key establishment in a group of users. This scheme, though primarily not intended for WSNs, was later used for key establishment in WSN. A symmetric polynomial-based scheme was proposed by Blundo et al. in [5]. Key predistribution schemes based on combinatorial design can be found in [6-12].

Combinatorial designs have been extensively used in deterministic key management. Mitchel and Piper [13] first used this in key distribution. In combinatorial design-based key distribution, a set system is used. The elements of the set system are regarded as the keys. A block is regarded as the key ring of a node. Çamptepe and Yener [6,7] were first to use combinatorial designs for key predistribution in sensor networks. They used projective geometry and generalized quadrangles. Lee and Stinson [8,9] used transversal designs for key distribution. Chakrabarti et al. [11] proposed a hybrid key predistribution scheme by randomly merging the blocks of the transversal design proposed by Lee and Stinson. Their merging technique

*Correspondence: samiran_r@isical.ac.in
Applied Statistics Unit, Indian Statistical Institute, 203 B T Road, Kolkata, WB 700108, India

enhances the resiliency of the key predistribution scheme of Lee and Stinson. Three designs were used by Dong et al. [14]. They also proposed a class of key predistribution scheme based on orthogonal array [15]. Blackburn et al. [16] proposed Costas arrays and distinct difference configuration. Product construction was used by [17]. The scheme is based on the product of key distribution scheme and set systems. They deduce the conditions of the set systems that provide optimum connectivity and resiliency of the network. Ruj and Roy proposed several schemes using partially balanced design, transversal design, and Reed-Solomon codes [10,18,19].

Key predistribution in wireless sensor networks using deployment knowledge was first studied by Liu and Ning [3]. They proposed two predistribution schemes both of which took advantage of the deployment knowledge of sensor nodes. The first scheme called the closest pairwise scheme was a modification of the pairwise key predistribution scheme. The second predistribution scheme uses the polynomial-based key predistribution scheme of Blundo et al. [5].

Several research works followed, e.g., [18,20-27]. In Du et al. scheme [20,21], the sensors are deployed in groups at a single point of deployment. The probability density function of the ultimate position of all sensors in a group are the same. They used multiple space Blom scheme [4] for key predistribution.

Yu and Guan [28,29] studied key predistribution schemes using deployment knowledge and compared the effect of deployment on triangular, hexagonal, and square grids. Huang et al. [24,25] proposed a grid-group-based key predistribution scheme. These schemes are perfectly secure to selective and random node capture attack. Here, the deployment area is divided into smaller rectangular zones of the same size. Every rectangular area contains equal number of sensors deployed uniformly in that zone. The keys in the sensors are deployed following multiple space Blom scheme similar to Du et al. scheme [20]. Each sensor node chooses keys from two key spaces such that no more than c sensors are chosen from the same key space, thus eliminating the possibility of node capture attacks. In [23], Zhou et al. discussed a key predistribution scheme where sensor nodes are mobile. There are static sensor which are deployed in groups. There are mobile collectors which are used to collect and aggregate sensor data and forward to the base station. The mobility of collectors enhance the data consistency.

Ruj and Roy [18] proposed a key predistribution for grid-group-based deployment. In this scheme, the deployment area is divided into smaller square regions. There are n^2 such smaller regions. There are two types of nodes viz. common nodes and agents. Their scheme offers full connectivity between the set of agents of the regions within the communication range.

Bag proposed a key predistribution scheme using the deployment knowledge in [30]. Here, the author considered a three-dimensional deployment zone where the sensor nodes are deployed not only along the length and breadth of the deployment zone but also along the height of the deployment zone.

In this paper, we propose a key predistribution scheme for homogeneous wireless sensor networks using the scheme of Blom [4] as well as symmetric balanced incomplete block design (SBIBD). The main advantage of using this scheme for key predistribution is that for this scheme, the adversary needs to capture large number of nodes in order to compromise all the keys in an uncompromised node. In other words, in order to disconnect an uncaptured node from all other nodes, the adversary needs to capture many more nodes than the other standard schemes.

Then, we use this new key predistribution scheme in a grid-group deployment of sensor nodes. A grid-group deployment refers to such a deployment where the entire deployment zone is broken into smaller two-dimensional square regions giving rise to an $n \times n$ grid-group structure. Equal number of sensor nodes are deployed in each of the smaller square regions of the deployment zone. Sensor nodes deployed inside one smaller square region forms a group. Sensor nodes within the same group communicate more frequently than a pair of nodes falling in two different groups. This is driven by the fact that sensor nodes in proximity to each other communicate more frequently than distant nodes. Sensor nodes deployed in this fashion grid form a heterogeneous network. This type of deployment scheme is applied in battlefields where sensors belonging to a compromised zone need to be completely disconnected from the rest of the network. Because if an adversary compromises an area, all the sensor nodes deployed in that area are considered to be captured.

This type of deployment is proposed by Liu and Ning [3,22]. There are two types of sensor nodes in this heterogeneous network. They mainly differ in resource. One type of nodes have a low amount of storage capacity, power, and computational power, and the other type of nodes are richer in the amount of computational resources that they posses. We shall use the name 'supernode' for the nodes which are more powerful than common nodes. Common sensors belonging to one region contain a set of keys that are completely disjoint from the sensors in some other region. This ensures that even if one region is totally disconnected, the other regions are not affected. For each sensor node, the keys are preloaded in such a way that all the nodes belonging to a particular square region (group) can communicate with each other directly. Sensor nodes belonging to different square regions (group) communicate through two or more supernodes.

Our general key predistribution scheme offers better resiliency than the schemes in [4,6,7]. For example, in key predistribution scheme by Blom [4], the adversary can compromise all the keys of the entire WSN merely by capturing c nodes, where c is the security parameter of the design. However, in our scheme, the adversary can only compromise few links by capturing c nodes. Our scheme also offers better resiliency than [6,7] in terms of the number of links that get exposed when some nodes are compromised. In both key predistribution schemes based on symmetric BIBD and generalized quadrangles in [6,7], the attacker can compromise many key links between pairs of uncaptured nodes by capturing a single node. However, in our scheme, the attacker needs to capture multiple nodes for compromising the key links between some pairs of nodes. We have compared our scheme with [18] and other similar schemes on the basis of fraction of links that gets exposed when some nodes get captured by the adversary. This is a well-known measure of the resiliency of a key predistribution scheme. Our scheme is shown to exhibit the best performance as far as the resiliency is concerned. The scheme of Ruj and Roy in [18] uses three times the number of supernodes we use in our scheme for full connectivity. Our scheme offers better resiliency using less number of supernodes.

2 Preliminaries

Here, we discuss some mathematical structures that we have used in our key predistribution scheme. Table 1 provides the meaning of different notations used in this section and in the next section.

2.1 Combinatorial design

A design [31] is a two tuple $(X, \mathcal{A})$ where X is a set of varieties, and $\mathcal{A}$ is a set of subsets of X:

$$\mathcal{A} = \{x : x \subseteq X\}$$

Table 1 Table of notations

Notations	
X	The set of varieties of the design
$\mathcal{A}$	The set of blocks of the design
$x_1, \ldots, x_v$	The varieties of X
$B_1, \ldots, B_b$	Blocks of $\mathcal{A}$
$GF(q)$	The finite field of q elements
α	The primitive element of $GF(q)$
G	A $c \times r$ matrix as defined below
t	The total number of nodes in deployment
p	A prime power where $t \leq p^2 + p + 1$
$\mathcal{N}$	The set of nodes in deployment
f	One to one map $\mathcal{N} \rightarrow \mathcal{A}$
D_i	$c \times c$ symmetric matrices over $GF(q)$ for $i = 1, 2, \ldots, v$

A (v, b, r, k, λ)-BIBD is a design satisfying these properties:

1. $|X| = v$.
2. $|\mathcal{A}| = b$.
3. $\forall B \in \mathcal{A}, |B| = k$.
4. $\forall x \in X, |\{B : B \in \mathcal{A}, x \in B\}| = r$.
5. $\forall x, y \in X, x \neq y, |\{B : B \in \mathcal{A},\ x, y \in B\}| = \lambda$.

A (v, b, r, k, λ)-BIBD, where $v = b$ is called a symmetric BIBD or SBIBD.

It can be shown that in a symmetric BIBD, $k = r$ [31].

A $(n^2 + n + 1, n + 1, 1)$-BIBD with $n \geq 2$ is called a projective plane of order n. It can be proven (Theorem 2.10, [31]) that for every prime power $q \geq 2$, there exists a symmetric $(q^2 + q + 1, q + 1, 1)$-BIBD i.e., a projective plane of order q.

2.1.1 Construction of SBIBD

Çamptepe and Yener used mutually orthogonal Latin squares in constructing the key predistribution scheme of [6]. Another construction of the same scheme can be found in [32]. Let $V_3(q)$ be the set of a three-dimensional vector space over a finite field F_q of q elements. A projective geometry $PG(2, q)$ over a finite field F_q is defined like the following:

- The points are given by the one-dimensional subspaces of $V_3(q)$.
- The lines are given by the two-dimensional subspaces of $V_3(q)$.
- A point belongs to a line if the corresponding one-dimensional subspace of the point is contained in the two-dimensional subspace corresponding to the line.
- Two lines are incident to each other iff the intersection of the corresponding two-dimensional subspaces of them is a nonempty one-dimensional subspace.

It can be shown that there are $(q^3-1)/(q-1)$ or q^2+q+1 number of distinct subspaces of dimension one of $V_3(q)$ [32]. Similarly, the number of distinct subspaces of dimension two of $V_3(q)$ is also $q^2 + q + 1$. Each two-dimensional subspace contains $q + 1$ distinct one-dimensional subspaces. The intersection of two-dimensional subspaces is a one-dimensional subspace of $V_3(q)$. So, the number of points and lines in $PG(2, q)$ is q^2+q+1. Every line contains $q + 1$ number of points. So, taking points as varieties and lines as block $PG(2, q)$ is a symmetric $(q^2 + q + 1, q + 1, 1)$ BIBD.

Since the lines of $PG(2, q)$ are two-dimensional subspaces of $V_3(q)$, we can represent each block by the basis of the subspaces they correspond to. The basis of a two-dimensional subspace of $V_3(q)$ contains exactly two elements. So, each block in $PG(2, q)$ will be identified by two elements of $V_3(q)$.

Similarly, the points of $PG(2,q)$ are one-dimensional subspaces of $V_3(q)$. So, every variety of $(q^2+q+1, q+1, 1)$ SBIBD can be represented by the basis of the one-dimensional subspace it belongs to.

Let $L_1 = \{(1,s,t) : s,t \in GF(q)\}$

$L_2 = \{(0,1,s) : s \in GF(q)\}$

$L_3 = \{(0,0,1)\}$

Let, $\mathcal{S} = L_1 \cup L_2 \cup L_3$.

$|\mathcal{S}| = q^2 + q + 1$.

It can be shown that each element of $\mathcal{S}$ is a basis of a distinct one-dimensional subspace of $V_3(q)$. Throughout this article, we shall represent the q^2+q+1 number of varieties of the $(q^2+q+1, q+1, 1)$ SBIBD by the elements of $\mathcal{S}$.

2.1.2 Shared variety discovery of $(q^2+q+1, q+1, 1)$ SBIBD

Any two blocks of a symmetric $(q^2+q+1, q+1, 1)$ BIBD do share one and unique variety. Given a $(q^2+q+1, q+1, 1)$ SBIBD, Algorithm 1 finds the common variety of two blocks of the design. This algorithm uses the basis of the nullspace of $A.x = 0$. This basis can be computed using Gauss-Jordan elimination method [33,34] in a constant time. Therefore, the runtime of Algorithm 1 is $O(1)$.

Algorithm 1 Computing the shared variety between two blocks of $(q^2+q+1, q+1, 1)$ SBIBD.

Require: Basis of block 1 $\{(a_1,b_1,c_1),(a_2,b_2,c_2)\}$.
Basis of block 2 $\{(a'_1,b'_1,c'_1),(a'_2,b'_2,c'_2)\}$.

Ensure: Find the identifier of the shared variety of the two blocks.

$$A = \begin{bmatrix} a_1 & a_2 & -a'_1 & -a'_2 \\ b_1 & b_2 & -b'_1 & -b'_2 \\ c_1 & c_2 & -c'_1 & -c'_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Find the basis of the nullspace $A.x = 0$.
Let this basis be given by $(\beta_1, \beta_2, \beta_3, \beta_4)$.
$a = a_1\beta_1 + a_2\beta_2$
$b = b_1\beta_1 + b_2\beta_2$
$c = c_1\beta_1 + c_2\beta_2$
if $a \neq 0$ **then**
  The identifier of the common variety is $(1, a^{-1}b, a^{-1}c)$.
else
  if $b \neq 0$ **then**
    The identifier of the common variety is $(0, 1, b^{-1}c)$.
  else
    The identifier of the common variety is $(0,0,1)$.
  end if
end if

2.2 Key predistribution using combinatorial design

Once we have a (v,b,r,k,λ)-design $(X,\mathcal{A})$, we can map it to a key predistribution scheme in the following way:

Let $\mathcal{K}$ be a set of v keys.
$\mathcal{N}$ be a set of b nodes in the WSN.
Let $\mathcal{A} = \{B_1, B_2, \ldots, B_b\}$ be the blocks of the design.
Let $f : \mathcal{K} \to X$ be a map and $g : \mathcal{N} \to \mathcal{A}$ be another map.
For each $B_i \in \mathcal{A}, i = 1,2,\ldots,b$ and
$\forall a_j \in X, j = 1,2,\ldots,v$ if $a_j \in B_i$ and both $f^{-1}(a_j)$ and $g^{-1}(B_i)$ exist, load key $f(a_j)$ into node $g(B_i)$.

In plain language, what we do here is to use varieties as keys and blocks as node. A node corresponding to a block contains all the keys corresponding to the varieties that the particular block contains. Two nodes will have a common key if and only if the corresponding blocks do share at least one common variety. Again, the number of keys in a node will be equal to the number of varieties in a block that corresponds to the node.

2.3 Blom's scheme

Blom [4] proposed a scheme for key predistribution where the members of a group can establish pairwise keys. Let N be the size of the network. The distribution server first chooses a $c \times N$ matrix G over a finite field $GF(q)$. The matrix G is considered to be a public information. Now, the distribution server constructs a $c \times c$ symmetric matrix D over $GF(q)$. This matrix is a private information of the system. Now, the server computes the $c \times N$ matrix A, where $A = (DG)^T$, T being the transposition operator. Now, $AG = (DG)^T G = G^T D^T G = G^T DG = G^T A^T = (AG)^T$.

Thus, AG is a symmetric matrix. Let $K = AG$, we know that $K_{ij} = K_{ji}$, where K_{ij} is the element in K located in the ith row and jth column. K_{ij} (or K_{ji}) is the pairwise key between node U_i and node U_j. To carry out the above computation, nodes U_i and U_j should be able to compute K_{ij} and K_{ji}, respectively. This can be easily achieved using the following key predistribution scheme, for $w = 1,2,\ldots,N$,

- Store the wth row of matrix A in node U_w.
- Store the wth column of matrix G in node U_w.

Now, if two nodes (say U_x and U_y) want to communicate, they need to establish a common key. Node U_x has row x of A and column x of G. Node U_y has row y of A and column y of G. Now , they can establish a pairwise key this way:

- Node U_x and U_y exchange column x and column y of matrix G, respectively.
- Node U_x calculates K_{xy} = (row x of A). (column y of G).
- Node U_y calculates K_{yx} = (row y of A). (column x of G).

The matrix G is a public information. Therefore, the rows of G could be sent without encryption. Since K is a symmetric matrix, $K_{xy} = K_{yx}$. Hence, K_{xy} can be used as the common key between the two nodes.

2.3.1 c-secure property

It has been proved that the above scheme is c-secure [4], i.e., if any $c+1$ columns of G are linearly independent; then, no member other than U_x and U_y can compute K_{xy} or K_{yx} if no more than c members are compromised.

2.3.2 A construction for matrix G

We note that any $c+1$ columns of G [35] must be linearly independent in order to achieve the c-secure property. Let α be a primitive element of a finite field $GF(q)$ where q is a prime power.

A feasible G can be designed as follows [36]:

$$G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^N \\ \alpha^2 & (\alpha^2)^2 & (\alpha^3)^2 & \cdots & (\alpha^N)^2 \\ \alpha^3 & (\alpha^2)^3 & (\alpha^3)^3 & \cdots & (\alpha^N)^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha^{c-1} & (\alpha^2)^{c-1} & (\alpha^3)^{c-1} & \cdots & (\alpha^N)^{c-1} \end{bmatrix}$$

It is well known that $\alpha^i \neq \alpha^j$ if $i \neq j$ (this is a property of primitive elements). Since G is a Vandermonde matrix, it can be shown that any $c+1$ columns of G are linearly independent when $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^N$ are all distinct. In practice, G can be generated by the primitive element α of $GF(q)$. Therefore, the wth column of G is stored at node U_w; it is only required to store the seed α^w, and any node can regenerate the column given the seed.

2.4 Threat model

Wireless sensor nodes are deployed in unattended environment often in an area under the control of adversaries. Thus, the sensor nodes that gather and communicate sensitive information are vulnerable to attacks. An active adversary can physically capture a number of nodes, and it can get to know the stored keys into them. These keys can thereafter be used by the adversary to decrypt messages communicated across sensor nodes. We shall discuss two types of attacks to our proposed scheme.

2.4.1 Random node capture

In this type of attack, the adversary randomly captures nodes from the deployment zone and exposes the keys loaded into them.

2.4.2 Selective node capture

This attack was first introduced in [37]. An active attacker is in attempt to obtain a set T of keys. For achieving this, the attacker is compromising sensor nodes. It has already obtained a set of keys S this way, where $S \subset T$. For each node s in the WSN, the random variable $G(s)$ is equal to the number of keys belonging to $T \setminus S$; the attacker gains by compromising s nodes. At each step of the attack sequence, the next sensor to be tampered with is sensor s, where s maximizes $E[\, G(s)|I(s)]$, the expectation of the key information gain $G(s)$ given the information $I(s)$ that the attacker knows on sensor s's key ring.

3 Proposed scheme

3.1 Key predistribution in the group

Here, our aim is to design a key predistribution scheme for a sensor network consisting $\mathcal{N}$ nodes where $\mathcal{N} \leq p^2+p+1$ where p is a prime number.

We use the scheme in [6,7] by Çamtepe and Yener and Blom's scheme [4]. This scheme is based on symmetric design (Section 2) . They used a symmetric $(p^2+p+1, p+1, 1)$ design to build a key predistribution scheme for WSN.

We shall be using a $(p^2+p+1, p+1, 1)$ -symmetric balanced incomplete block design $(X, \mathcal{A})$. Here, $X = \{x_1, x_2, \ldots, x_v\}, v = p^2+p+1$. $\mathcal{A} = \{B : B = \{x_{j_1}, x_{j_2}, \ldots, x_{j_{p+1}}\}, j_1, j_2, \ldots, j_{p+1} \in \{1, 2, \ldots, v\}, j_m \neq j_n, 1 \leq m, n \leq p+1\}$. $|\mathcal{A}| = p^2+p+1$. Here, B_is are the individual blocks for all $i \in \{1, 2, \ldots, p^2+p+1\}$. $|B_i| = p+1, \forall i \in \{1, 2, \ldots, p^2+p+1\}$.

3.1.1 The scheme

Definition 1. *For any node $n_i \in \mathcal{N}$, and a variety $x_l \in X$ and a block $B_d \in \mathcal{A}$, $POS(B_d, x_l)$ is an integer taking values from the set $\{1, 2, \ldots, k\}$, where $f(n_i) = B_d$ and $x_l \in B_d$. The node n_i stores the values of $POS(B_d, x_l), \forall x_l \in B_d$.*

Since, $|B_d| = k, \forall B_d \in \mathcal{A}$, so each node stores k number of $POS(*, *)$ values.

Definition 2. *f is a one-to-one map from the set of nodes of the sensor network to the blocks of the symmetric $(p^2+p+1, p+1, 1)$ design. In addition to that, we assume that f^{-1} can be computed in constant time.*

It can be noted that the nodes can be identified by the identifier of the blocks they correspond to. Therefore, one example of the function f is the identity mapping if $\mathcal{N} \subseteq \mathcal{A}$.

The total number of nodes in deployment be $t = |\mathcal{N}|$. Choose a prime power p such that $t \leq p^2+p+1$. Now, design a symmetric $(p^2+p+1, p+1, 1)$ BIBD using Algorithm 1 of [7]. Comparing a (v, b, r, k, λ)-design to this symmetric $(p^2+p+1, p+1, 1)$-design, we get $v = b = p^2+p+1, k = r = p+1$ and $\lambda = 1$. The varieties of the design are denoted by $x_1, x_2, \ldots, x_{p^2+p+1}$ and the blocks as $B_1, B_2, \ldots, B_{p^2+p+1}$. We shall design our key predistribution scheme in nodes using this symmetric $(p^2+p+1, p+1, 1)$-design. Let the security parameter be c as in Section 2.3. We shall later discuss on a feasible

value the integer c. Now, compute p^2+p+1 symmetric $c \times c$ matrices $D_1, D_2, \ldots, D_{p^2+p+1}$ over a finite field $GF(q)$. Now, construct a $c \times r$ matrix G using the method described in 2.3 i.e. if α is a primitive element of $GF(q)$, compute:

$$G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^r \\ \alpha^2 & (\alpha^2)^2 & (\alpha^3)^2 & \cdots & (\alpha^r)^2 \\ \alpha^3 & (\alpha^2)^3 & (\alpha^3)^3 & \cdots & (\alpha^r)^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha^{c-1} & (\alpha^2)^{c-1} & (\alpha^3)^{c-1} & \cdots & (\alpha^r)^{c-1} \end{bmatrix}$$

Algorithm 2 maps a (v, b, r, k, λ) design $(X, \mathcal{A})$ of Section 2.1 into a key predistribution scheme. Let $\mathcal{N} = \{n_1, n_2, \ldots, n_t\}$ be the set of nodes in the WSN. We can design a key predistribution in these nodes using Algorithm 2 and taking $v = b = p^2+p+1, r = p+1$. In Algorithm 2, we take $v = p^2+p+1$ many different key spaces of the Blom scheme [4]. We compute one $c \times r$ public matrix G and a set of v many $c \times c$ secret symmetric matrix $D_i, i \in \{1, 2, \ldots, v\}$. Thus, we can compute v many A matrices like this : $A_i = (D_i \dot{G})^T$. Hence, there are v many distinct key spaces of Blom scheme. Now, we can have a key distribution scheme by considering each of the v key space as a variety of the $(p^2+p+1, p+1, 1)$- SBIBD, where each block of the SBIBD corresponds to a node of the WSN. Since a block of a $(p^2+p+1, p+1, 1)$- SBIBD contains $p+1$ many varieties, every node will have its key share from exactly $p+1$ many key spaces.

Algorithm 2 Algorithm for key predistribution in nodes.

Require: A combinatorial design $(X, \mathcal{A})$ where
$X = \{x_1, x_2, \ldots, x_v\}$,
$\mathcal{A} = \{B_1, B_2, \ldots, B_b\}$,
$\mathcal{N} = \{n_1, n_2, \ldots, n_t\}$,
$f : \mathcal{N} \to \mathcal{A}$ is a one-one map,
A $c \times r$ Matrix G,
v number of $c \times c$ Matrices $D_1, D_2, \ldots, D_v$.

Ensure: A key predistribution in sensor nodes of $\mathcal{N}$.

for all $x_j \in X, 1 \le j \le v$ **do**
  Find ordered set $S = \{B_{j_1}, B_{j_2}, \ldots, B_{j_r}\}$ be such that $B_{j_k} \in \mathcal{A}, x_j \in B_{j_k}; \forall k \in \{1, 2, \ldots, r\}$; $B_{j_k} \ne B_{j_l}, 1 \le k, l \le r$ and $\forall B \in \mathcal{A} \setminus S, x_j \notin B$.
  Compute $A_j = (D_j.G)^T$
  for all $i \in \{1, 2, \ldots, r\}$ **do**
    if $f^{-1}(B_{j_i})$ exists **then**
      Store the ith row of matrix A_j in node $f^{-1}(B_{j_i})$
      Store the 2nd row of G in node $f^{-1}(B_{j_i})$
      In node $f^{-1}(B_{j_i})$, store $POS(B_{j_i}, x_j) = i$.
    end if
  end for
end for

3.1.2 Memory requirement

It is easy to see that one node n_h contains one row from each matrix of the set M_h where $M_h \subset \{A_1, A_2, \ldots, A_v\}$ where $|M_h| = k$. The dimension of each row is c. Also, the node contains row 2 of matrix G which is $(\alpha, \alpha^2, \ldots, \alpha^r)$. It can be seen that for $(p^2+p+1, p+1, 1)$ SBIBD $r = k$. Again, a node n_i stores $POS(f(n_i), x_i)$ for $i \in \mathbb{V}, \mathbb{V} \subset \{1, 2, \ldots, v\}, |\mathbb{V}| = k$. So, the overhead on each node is $O(kc + r + k)$. For most of the cases, c is a small constant. In this design $k = p+1$. Therefore, the memory overhead is $O(p)$ or $O(\sqrt{|\mathcal{N}|})$.

3.1.3 Shared key discovery between two nodes

Two nodes wishing to communicate securely need to agree upon a secret key. In the scheme discussed in Section 3.1.1, any two nodes can surely compute a shared key. We provide an algorithm that takes all arguments of Algorithm 2 and finds a shared key between two nodes. In addition, the algorithm takes two nodes as input and finds a common key shared by both of them.

The most costly computation of Algorithm 3 is at step 3. This step reduces in finding all the blocks of a design that contains a particular variety. This can be found using a different construction of symmetric BIBD as discussed in Section 8.4 of [32].

Algorithm 3 Algorithm to compute common key between node n_i an n_j.

Require: Combinatorial design $(X, \mathcal{A})$ used in Algorithm 2 where
$X = \{x_1, x_2, \ldots, x_v\}$,
$\mathcal{A} = \{B_1, B_2, \ldots, B_b\}$,
$\mathcal{N} = \{n_1, n_2, \ldots, n_t\}$,
$f : \mathcal{N} \to \mathcal{A}$ is a one-one map,
A $c \times r$ Matrix G,
v number of $c \times c$ Matrices $D_1, D_2, \ldots, D_v$.

Ensure: Compute the common key between node n_i and n_j

1) Let, $B_y = f(n_i), B_z = f(n_j)$
2) Compute $x_m \in X, m \in \{1, 2, \ldots, v\}$ such that $x_m \in B_y \cap B_z$
3) Find$u = POS(B_y, x_m)$ and $w = POS(B_z, x_m)$
4) Compute wth column of matrix G from $(\alpha, \alpha^2, \ldots, \alpha^r)$.
5) $K_{n_i, n_j} = (u$th row of matrix $A_m).(w$th column of matrix G)

Time complexity of Algorithm 3 The first step reduces in inverting the node ids. We assumed that f is invertible in constant time. So, the first step can be done in time $O(1)$. The second step computes a common variety

belonging to two different blocks in the design used in Algorithm 2. Note that in a $(p^2+p+1, p+1, 1)$-SBIBD, any two blocks will share a unique common variety. Computing such a variety in a $(p^2+p+1, p+1, 1)$-SBIBD is equivalent to computing a basis of the intersection of two-dimensional subspaces. This can be done in constant time using the Algorithm 1. The third step is a lookup of memory and is, too, of time complexity $O(1)$ if the items are stored in an indexed table. In the fourth step, the wth column of matrix G is calculated which is given by $(1, \alpha^w, (\alpha^w)^2, (\alpha^w)^3, \ldots, (\alpha^w)^{c-1})'$. Since the nodes store α^i for each $i = 1, 2, 3, \ldots, r$ and c is a constant, so computing the wth column of matrix G requires $O(1)$ computation. Finally, the fifth step can also be done in constant time since the vectors are of constant dimension. Therefore, the overall runtime of Algorithm 3 is $O(1)$.

Note that node n_i stores the value $u = POS(B_y, x_m)$ in the Algorithm 3, and node n_j stores the value of $w = POS(B_z, x_m)$. However, for computing the shared key, both the nodes need the values of u and w. So, the two nodes must exchange the values of u and w which will incur an additional communication cost of $O(1)$. To avoid this, every node can store the values of $POS(*, *)$ for other nodes. For example node $n_i = f^{-1}(B_y)$ needs to store the values of $POS(B_e, x_l) : 1 \leq e \leq v, e \neq y, x_l = B_e \bigcap B_y$. This will require a memory overhead of $O(\mathcal{N})$.

3.2 Proof of correctness of algorithms

Here, we establish the correctness of Algorithm 2 and Algorithm 3. It will be sufficient to show that after deployment, a pair of distinct nodes n_i and $n_j, 1 \leq i, j \leq v$ will be able to compute their common key $K_{n_i n_j} = K_{n_j n_i}$ using the shared key discovery method of Algorithm 3. According to Algorithm 3, both node n_i and node n_j will compute the blocks $B_y = f(n_i)$ and $B_z = f(n_j)$. Now, they can find the common element $x_m \in B_y \bigcap B_z : 1 \leq m \leq v$ using Algorithm 1. Now, node n_i will compute $u = POS(B_y, x_m)$. Similarly, node n_j will calculate $w = POS(B_z, x_m)$. Node n_i and n_j will exchange the values u and v. Node n_i will compute the wth column of matrix G from $(\alpha, \alpha^2, \ldots, \alpha^r)$ stored in it. Similarly, node n_j will calculate the uth column of matrix G from $(\alpha, \alpha^2, \alpha^r)$ stored in it. From Algorithm 1, we can see that node n_i and n_j have got the uth and wth row of matrix $A_m = (D_m.G)^T$. Hence, node n_i can compute $K_{uw} =$ (uth row of matrix A_m).(wth column of matrix G. Node n_j will compute $K_{wu} =$ (wth row of matrix A_m).(uth column of matrix G in a similar way. Since $A_m.G$ is a symmetric matrix, $K_{uw} = K_{wu} = K_{n_i n_j}$. Hence, the two nodes will end up computing the same key using Algorithm 3. Therefore, the Algorithm 2 and 3 are correct. It can be noted that any row of matrix $A_k, 1 \leq k \leq v$ is contained only in exactly one node according to Algorithm 2. So, only node n_i contains the uth row of A_m and only node n_j contains the wth row of A_m. Hence, no other node can compute the common key $K_{n_i n_j}$.

4 Performance analysis of proposed scheme

In this section, we shall investigate the security aspects of the proposed scheme. As discussed in Section 1, sensor nodes are deployed in unattended environment often in area controlled by an adversary. So, an active adversary can compromise one or more sensor nodes of the deployment zone. If the sensor nodes are not tamper proof, the adversary can extract sensitive information from the set of sensor nodes compromised by the adversary and can use those informations to overhear the conversation between active sensor nodes.

Lemma 3. *For the proposed scheme, let* $\mathbf{S}$ *be the set of compromised sensor nodes. Let,* $f(\mathbf{S}) = \{f(n) : n \in \mathbf{S}\}$. *Two uncompromised nodes* n_1 *and* n_2 *will have an uncompromised link between them if and only if* $|\{B : B \in f(\mathbf{S}) \& x \in B\}| \leq c-1$, *where* $x = B_1 \cap B_2$ *and* $f(n_1) = B_1, f(n_2) = B_2$.

Proof. Follows from the fact that c is the security parameter of the scheme in Section 2.3.

Let, $x = x_\kappa$, where $\kappa \in \{1, 2, \ldots, v\}$. Then by Algorithm 2 and 2.3, it can be said that if the matrix A_κ can be compromised, then the common key between node n_1 and n_2 can be computed. This can only be possible if and only if any c number of rows of the matrix A_κ are compromised. Let $\psi = \{n : n \in \mathcal{N} \& x_\kappa \in f(n)\}$. Hence, the nodes in ψ contain one distinct row of A_κ each. So, successful computation of the shared key is possible if and only if $|\mathbf{S} \cap \psi| \geq c$. In other words, the common key between the two nodes n_1 and n_2 will remain active if and only if $|\{B : B \in f(\mathbf{S}) \& x \in B\}| \leq c - 1$. □

Proposition 4. *Let the total number of nodes be N and the security parameter be c. If s number of nodes are compromised and* $s \geq c$, *the probability that two uncompromised nodes will have an uncompromised link is given by* $\frac{\sum_{e=0}^{c-1} \binom{k-2}{e}\binom{N-k}{s-e}}{\binom{N-2}{s}}$.

Proof. Let C denote the event that the two nodes will share an uncompromised link. Let the two nodes be given by n_1 and n_2. Let, $f(n_1) = B_1$ and $f(n_2) = B_2$, where $B_1, B_2 \in \mathcal{A}$. There must be a unique $x_i \in X$ such that $\{x_i\} = B_1 \cap B_2$. Again, let the set of compromised nodes be $\mathbf{S}$, where $|\mathbf{S}| = s$. The adversary cannot compute the shared key between n_1 and n_2 iff $|\{B : B \in f(\mathbf{S}) \& x_i \in B\}| \leq c - 1$. In a symmetric (v, k, λ) design, there are k number of blocks containing a particular variety. So, for any particular variety $x_i \in X$, $|\{B : B \in \mathcal{A} \& x_i \in B\}| = k$. Again, $B_1, B_2 \in \{B : B \in \mathcal{A} \& x_i \in B\}$. Therefore, $|\{B : B \in \mathcal{A} \& x_i \in B, B \neq B_1, B \neq B_2\}| = k - 2$.

$P(|\{B : B \in f(\mathbf{S})\&x_i \in B, n_1, n_2 \notin S\}| = e) = \frac{\binom{k-2}{e}\binom{N-k}{s-e}}{\binom{N-2}{s}}$.
$\therefore P(C) = \sum_{e=0}^{c-1} P(|\{B : B \in f(\mathbf{S})\&x_i \in B, n_1, n_2 \notin S\}| = e) = \frac{\sum_{e=0}^{c-1}\binom{k-2}{e}\binom{N-k}{s-e}}{\binom{N-2}{s}}$. □

We provide the values of $P(C)$ for different sets of parameters in Table 2. It can be seen that our scheme has high probability of existence of a live link between two uncaptured nodes even when large number of nodes are compromised. Here, p is the prime number of the symmetric balanced incomplete block design that is used in the scheme. c is the security parameter of Blom's scheme. s is the number of compromised nodes. Table 2 shows that this scheme has a high probability of existence of a key link between two nodes even when many nodes are compromised. Also, if p increases, the number of nodes increases and so does the probability of existence of a link between a pair of nodes.

4.1 Performance analysis in terms of known measures

We shall analyze the performance of our scheme in terms of two well-known measures viz. $E(s)$ and $V(s)$. These are the standard measures used for evaluating the resiliency of any key predistribution scheme.

Definition 5. *$E(s)$ is defined to be the ratio of the number of links exposed in the network when s number of nodes are compromised to the number of links present in the network before s number of nodes were compromised.*

Let, L be the total number of links in a network and l be the number of links exposed after s number of nodes are compromised.

then $E(s) = \frac{l}{L}$

Here, we will consider only the resiliency of the subnetwork consisting of nodes. $E(s)$ is the measure that shows the performance of the scheme in terms of it's resiliency against node captures. As defined above, $E(s)$ is the measure that shows the fraction of links that gets exposed when s number of nodes get compromised. So, the lesser the value of $E(s)$ is, the more resilient is the scheme to node capture attack.

Let $\mathbf{S}$ be the set of s sensor nodes. $\mathbf{S} \subseteq \mathcal{N}$. For two sensor nodes $n_i, n_j \in \mathcal{N}$, define

$$LNK(n_i, n_j) = \begin{cases} 0 & \text{if the adversary can compute the common key between node } n_i \text{ and } n_j \text{ using the information stored in nodes } n_\kappa, \kappa \in \mathbf{S} \\ 1 & \text{elsewhere} \end{cases}$$

From Lemma 3,

$$LNK(n_i, n_j) = \begin{cases} 0 & \text{if } |\{B : B \in f(\mathbf{S})\&x \in B\}| \geq c, \text{ where } x = B_1 \cap B_2 \text{ and } f(n_1) = B_1 \; f(n_2) = B_2 \\ 1 & \text{if } |\{B : B \in f(\mathbf{S})\&x \in B\}| \leq c-1, \text{ where } x = B_1 \cap B_2 \text{ and } f(n_1) = B_1 \; f(n_2) = B_2 \end{cases}$$

Let $\varphi(\mathbf{S}) = \frac{\sum_{i=1}^{t}\sum_{\substack{j=1 \\ j\neq i}}^{t} LNK(n_i,n_j)}{t(t-1)}$

Hence, $E(s) = EXP(\varphi(\mathbf{S}))$, where $EXP()$ is the expectation operator.

Theorem 6. *For our scheme with p^2+p+1 many nodes, $E(s) \leq \frac{c}{p^2+p+1}$ for $s \leq \frac{c(c+1)}{2}$*

Proof. The total number of nodes is $p^2 + p + 1$. That makes the number of links equal to $\binom{p^2+p+1}{2}$.

We take the attacker's point of view who would try to expose more links through compromising as less number of nodes as possible. In our design, a link can be exposed only if at least c number of nodes are compromised that contain one row of matrix A_h, each for some $h \in \{1, 2, \ldots, v\}$. If c number of rows are compromised, then the attacker would be able to reconstruct the matrix A_h. Since A is a $(p + 1) \times c$, the attacker would be able to compute the common keys between $\binom{p+1}{2}$ pair of nodes or in other words $\binom{p+1}{2}$ links would get exposed. Let, $n_0, n_1, \ldots, n_p$ be $p + 1$ nodes such that $x_i = \cap_{j=0}^{p} f(n_j)$ for any $x_i \in X, i \in \{1, 2, \ldots, v\}$. If any c of the nodes $n_0, n_1, \ldots, n_p$ is compromised by the adversary, then we would be able to reconstruct matrix A_i and hence, the links between nodes $n_0, n_1, \ldots, n_p$ will get exposed. So, the total number of exposed links will be $\binom{p+1}{2}$. Let the set of nodes compromised by the advisor for obtaining A_i be S. Hence, $|S| \geq c$. Since, the attacker's intention is to compromise as less number of nodes as possible, we can say, $|S| = c$. Again, the attacker would attempt to expose another set of $\binom{p+1}{2}$ links by compromising more nodes.

Table 2 Probability of existence of an active link between two uncompromised nodes in our scheme for different parameters

p	c	s	Probability of existence of link
37	4	34	0.998651
37	4	77	0.869753
47	4	77	0.930515
61	4	89	0.948484
67	5	89	0.991089
67	4	128	0.886519
61	5	110	0.970800
71	5	223	0.810936

Here, p is the prime number, s is the number of compromised nodes, and c is the security parameter of the scheme.

The attacker can do this through compromising another matrix $A_j, j \neq h, j \in \{1, 2, \ldots, v\}$. This time, the attacker needs to compromise $c-1$ nodes. First, an attacker selects a $j \neq h$ such that a node in S does contain a row A_j. Choosing such a j will ensure that the attacker will have to compromise $c-1$ more nodes. It can be proved that for any $j \neq h$, there is at most one node in S that contains a row of matrix A_j. So, the attacker would require to compromise $c-1$ additional nodes for exposing $\binom{p+1}{2}$ links. This way, it can be proved that the attacker would require to compromise $c-2$ nodes for exposing the next set of $\binom{p+1}{2}$ number of links and so on. This way, the attacker can compromise $c\binom{p+1}{2}$ number of links by capturing $c + (c-1) + (c-2) + \ldots + 1$ nodes or $\frac{c(c+1)}{2}$ nodes.

Hence, for $s \leq \frac{c(c+1)}{2}, E(s) \leq c\binom{p+1}{2}/\binom{p^2+p+1}{2}$ or, $E(s) \leq \frac{c}{p^2+p+1}$. □

Theorem 6 gives an upper bound of the extent of damage that occurs to the subnetwork consisting of nodes. Since $p^2 + p + 1 >> c$, so $E(s)$ is very close to zero or, in other words, the number of links that get exposed is small when less than $\frac{c(c+1)}{2}$ number of nodes are captured.

Lemma 7. *If a set of* **S** *sensor nodes get captured, then a node* $n_i \notin$ **S** *will get disconnected from the rest of the network if and only if* $\forall x \in f(n_i), |\{B : B \in f(\mathbf{S}), x \in B\}| \geq c$.

Proof. The proof follows from Lemma 3 and the c security property. □

Definition 8. *$V(s)$ is the fraction of nodes that get disconnected from the rest of the networks. Let m be the number of uncompromised nodes that get disconnected from the rest of the network of size N when s nodes are compromised, then $V(s) = \frac{m}{N-s-m}$.*

Theorem 9. $V(s) = 0, \forall s < (p+1)c$.

Proof. Let the attacker wants to disconnect a particular node $n_i, i \in \{1, 2, \ldots, v\}$ from the rest of the network. Let, S be the minimal set of nodes that the attacker needs to capture for disconnecting the first (uncompromised) node from the rest of the network. Let $B_j = f(n_i), j \in \{1, 2, \ldots, v\}$. Hence, $B_j \in X$. Let, $\{x_1, x_2, \ldots, x_{p+1}\} = B_j$. Let $\forall k \in \{1, 2, \ldots, p+1\}, C_k = \{B : B \in f(S) \& x_k \in B\}$. It can be seen that $f(S) = \cup_{k=1}^{p+1} C_k$.

We claim that $C_k \cap C'_k = \phi, k \neq k', 1 \leq k, k' \leq p+1$. If not, then suppose there exists a block $B_m \in C_k \cap C'_k$. Hence, $x_k, x'_k \in B_m$. So, $|B_m \cap B_j| \geq 2$. This is not possible since the design we used is a symmetric $(p^2+p+1, p+1, 1)$ design. So our assumption is wrong.

From the c-security property, we can say that $|C_k| = c \forall k \in \{1, 2, \ldots, p+1\}$. Hence, $|f(S)| = |S| = (p+1)c$. Hence the result. □

The performance of our scheme in terms of $V(s)$ for certain value of parameters is shown in Figure 1. It can be seen that the value of $V(s)$ in Figure 1 is in agreement with the result stated in Theorem 9.

4.2 Comparative study of the scheme

Here, we compare the resiliency of our proposed scheme with other existing schemes. Some well-known standard schemes are the basic scheme of Eschenauer and Gligor [1], Lee and Stinson's quadratic and linear scheme based on transversal design in [8,9,38], Çamptepe and Yener's

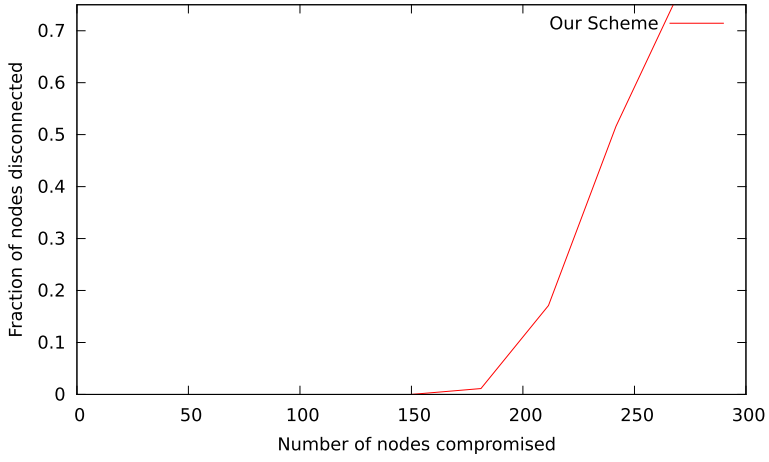

Figure 1 Graphical representation of the value of $V(s)$ with respect to the number of nodes compromised for our scheme. The parameters for this graph is p=29,c=4, and number of nodes=871.

scheme in [7], the scheme of Chakrabarti et al. [11], and partially balanced incomplete block design based scheme by Ruj and Roy in [10].

The scheme of Eschenauer and Gligor in [1] is a probabilistic key predistribution scheme. This scheme uses a pool of keys. Keys are drawn randomly from the key pool with replacement and are placed in the sensor nodes. All nodes are loaded with same number of keys. This scheme does not ensure the existence of a common key between a pair of nodes. This scheme is known as the basic scheme.

Lee and Stinson [8,38] used transversal design in key predistribution. They proposed two types of transversal design viz. linear and quadratic. In these schemes, a pair of nodes can have zero or one key in common. They used the following construction of a transversal design $TD(k,r)$ [8].

1. $X = \{(x,y) : 0 \leq x < k, 0 \leq y < r\}$.
2. $\forall i, G_i = \{(i,y) : 0 \leq y < r\}$.
3. $A = \{A_{i,j} : 0 \leq i < r \& 0 \leq j < r\}$.

They defined block $A_{i,j}$ by $A_{i,j} = (x, xi + j \mod r) : 0 \leq x < k, 0 \leq i,j < r$. Similarly for a quadratic scheme, they defined a block $A_{i,j,k}$ by $A_{i,j,k} = \left(x, xi^2 + xj + k \mod r\right) : 0 \leq x < k, 0 \leq i,j < r$.

Each block is assigned to a node. So, the linear Lee-Stinson's scheme supports r^2 nodes, and the quadratic scheme supports as many as p^3 nodes.

Çamtepe and Yener used symmetric balanced incomplete block design in [7]. A SBIBD is a $(p^2+p+1, p+1, 1)$ design where p is a prime number. They used projective geometry for constructing the SBIBD. This scheme ensures full connectivity between nodes. Each node in this scheme contains $p+1$ keys, and every key is contained in $p+1$ nodes.

Chakrabarti et al. [11] proposed a hybrid key predistribution scheme by merging the blocks in combinatorial designs. They considered the blocks constructed from the transversal design proposed by Lee and Stinson and randomly selected them and merged them to form the sensor nodes. Though this scheme increases the number of the keys per node, it improves the resiliency of the network. The probability that two nodes share a common key is also high. Thus, it has a better connectivity.

Ruj and Roy proposed two schemes for key predistribution in [10]. They used partially balanced incomplete block design. In the first scheme, the number of nodes as well as the number of keys are equal to $n(n-1)/2$ for some positive integer n. The number of keys in a node is equal to $2(n-2)$. The number of nodes containing the same key is also $2(n-2)$. They presented another design that augments the size of the network, keeping the same number of keys in each node. The keys in the key pool also remain the same. They showed that network size can be increased in steps, keeping the same number of keys per node. However, to ensure that any pair of nodes can communicate directly, we cannot go on adding nodes in this scheme.

We have defined $E(s)$ in Section 4.1. $E(s)$ is the best measure of resiliency of any key predistribution scheme. A key predistribution scheme for which the value of $E(s)$ is lower offers better resiliency against node capture. So, a key predistribution scheme having low value of $E(s)$ for different values of captured nodes can withstand key compromise. Figure 2 shows a comparison between our scheme with these schemes in terms of $E(s)$. We measured the resiliency of the key predistribution schemes by means

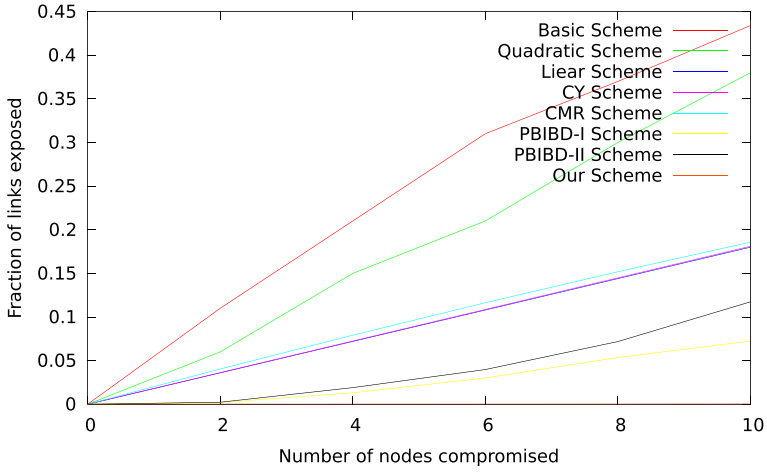

Figure 2 Graphical comparison of fraction of links exposed. With respect to the number of nodes compromised for our scheme and other schemes. The parameters for this comparison can be found in Table 3. The line corresponding to the performance of our scheme almost touches the horizontal axis and hence can hardly be seen.

of simulation. The parameters of different key predistribution schemes and the number of nodes in the WSN are given in Table 3. We have chosen nearly equal sizes of networks for different schemes in consideration. The other parameters are chosen depending upon the network size and the system models so that the key predistribution schemes exhibit optimal performance. N is the total number of nodes in the network, and k is the number of keys per node. The value of k depends upon the other parameters of the network which in turn depend upon the network size. The last column of Table 3 shows whether the key predistribution scheme ensures full connectivity among the nodes or not. We used C program to evaluate the values of $E(s)$ for different values of s for all the schemes mentioned above. We compiled the source using GNU C compiler GCC 4.5.4. We considered random node capture by the adversary. In Figure 2, the line corresponding to the performance of our scheme almost touches the x-axis throughout the range. Hence, it can be inferred that less number of links get exposed in our scheme as compared to other schemes when same number of nodes are captured by the adversary. In other words, our scheme offers better performance than all the other schemes in terms of $E(s)$. The reason why our scheme excels in performance can be inferred from Lemma 3. Lemma 3 says that in order to compromise the links between any two nodes, the adversary is required to compromise at least c (c is the security parameter) nodes having information from the same key space as the two nodes. However, in other schemes, the same thing can be done by capturing a single node. So, even if the number of captured nodes is high enough, the value of $E(s)$ can be very low in our scheme. This fact is corroborated by the performance of our scheme as shown in Figure 2.

5 New grid-group deployment-based design

We shall use our proposed key predistribution scheme in developing a key predistribution scheme for grid-group deployment. As mentioned earlier in Section 1, a grid-group deployment refers to such deployment where the entire network is broken into smaller regions called groups. The sensor nodes belonging to one group could be deemed as a mini-WSN where the sensors of a certain group communicates among themselves more frequently than with sensors of different groups. We propose a key predistribution scheme for a WSN where the network is divided into a $N \times N$ square grid. Each group in this group has got identical number of sensors.

Table 3 Schemes with parameters that we choose for our comparisons and connectivity

Scheme	N	k	Full connectivity
Basic [1]	2,415	136	No
Camtepe-Yener [6]	2,257	48	Yes
Linear [38]	2,209	30	No
Quadratic [38]	2,197	12	No
CMR [11]	2,550	28	No
PBIBD I [10]	2,415	136	Yes
PBIBD II [10]	2,450	96	Yes
Current scheme	2,257	48	Yes

N is the total number of nodes in the network, and k is the number of keys in a node.

5.1 The scheme

Let p be a prime number. Let $\mathcal{N} \leq p^2+p+1$ be the number of sensors in each group. The groups are denoted by the two tuple $(i,j), 0 \leq i,j \leq \mathcal{N}$. We shall denote the nodes of any group (i,j) as $n_{ij}^l, 0 \leq l \leq t-1$. We designate one node from each group as a supernode. This supernode has got more amount of resources than ordinary nodes in terms of memory, computational power, battery power, etc. This special node will be used for intergroup communication. The supernode of group (i,j) is denoted by $S_{i,j}$. It can be noted that a supernode $S_{i,j}$ of any group (i,j) does belong to the set $\{n_{ij}^l : 0 \leq l \leq t-1\}$. If a node $n_{i,j}^\alpha$ of group (i,j) wants to communicate with node $n_{i',j'}^\beta$ of group (i',j'), then the following steps are taken:

- Node $n_{i,j}^\alpha$ generates a random key $\mathbb{K}$.
- Node $n_{i,j}^\alpha$ send $\mathbb{K}$ to the supernode S_{ij}.
- S_{ij} passes $\mathbb{K}$ to $S_{i'j'}$.
- $S_{i'j'}$ sends $\mathbb{K}$ to node $n_{i',j'}^\beta$.

Now, the two nodes viz $n_{i,j}^\alpha$ and $n_{i',j'}^\beta$ can communicate using the key $\mathbb{K}$.

It can be noted that for accomplishing all the steps mentioned above, it is necessary to have:

1. Any two pair of nodes $n_{i,j}^\alpha$ and $n_{i,j}^{\alpha'}$ belonging to group (i,j) must be able to communicate securely $\forall \alpha \in \{0,1,2,\ldots,t-1\}$ and $0 \leq i,j \leq p-1$.
2. Any pair of supernodes $S_{i,j}$ and $S_{i',j'}$ belonging to two different groups (i,j) and (i',j') must be able to communicate securely where $0 \leq i,j,i',j' \leq p-1, (i,j) \neq (i',j')$.

We now state our key predistribution scheme in detail. From the above discussion, it is clear that we need to have two types of key predistribution. One type of key predistribution is for the nodes within each of the groups and the other for the supernodes belonging to distinct groups. For each of the N^2 groups, we use our key predistribution scheme discussed in Section 3 for key predistribution. However, we do use distinct key spaces for key predistribution in each of the groups. Hence, if all the nodes corresponding to one region get captured in the

hands of the adversary, the keys in sensor nodes in other groups remain unaffected. It should be kept in mind that a supernode belongs to the group corresponding to the square region they are deployed in. Hence, a supernode contains two types of keys, one that allows it to communicate securely with other nodes in the same group they belong to and the other that allows it to communicate with other supernodes belonging to different groups. Therefore, the key predistribution in the whole network looks like the following :

1. Key predistribution for each of the N^2 groups is done by using the scheme of Section 3 using exclusive key spaces for all the groups.
2. A separate key predistribution using the same scheme of Section 3 is done for all the supernodes belonging to all the groups.

We assume that it is hard to capture a supernode until the entire square region where the supernode is located is compromised. We have assumed that the nodes within the same square region communicate more frequently than the two nodes each belonging to a separate square region. Hence, one supernode per group is sufficient to handle the burden of intergroup communication.

5.2 Resiliency of the network

When it comes to the resiliency of the key predistribution scheme in a grid-group deployment of the sensor network, there are three types of resiliency:

- Intragroup resiliency : resiliency within a certain group.
- Resiliency of the interlinks : resiliency in the set of supernodes.
- Overall resiliency : resiliency of the entire network.

Within a group, the nodes work as a single WSN. Hence, the resiliency of the key predistribution is same as in Section 4. In this section, we study the resiliency of the interlinks in our key predistribution scheme. Here, too, similar to Section 4, we shall be using the standard measures for evaluating the resiliency of our scheme. The two measures we shall be using are $E'(s)$ and $V'(s)$.

Definition 10. *$E'(s)$ is defined to be the fraction of interlinks between groups that get exposed when s number of supernodes are captured by the adversary. In other words, $E'(s)$ is the ratio of the interlinks present in the grid after s many supernodes are captured to the number of interlinks present in the network before s many supernodes are captured.*

Let $\mathbb{S} = \{(i,j) : 0 \le i,j \le N-1\}$

$$\mathsf{K}_{(i,j)}(h,k) = \begin{cases} 1 & \text{if the common keybetween } S_{i,j} \text{and} S_{h,k} \\ & \text{exists} \\ 0 & \text{elsewhere} \end{cases}$$

Also, let for any group (i,j),

$$T(i,j) = \sum_{\substack{(i',j')\in\mathbb{S} \\ (i',j')\neq(i,j)}} \mathsf{K}_{(i,j)}(i',j')$$

It can be seen that in our design, all the supernodes have a common key between each other. Hence,
$T(i,j) = N^2 - 1 \qquad \forall (i,j) \in \mathbb{S}.$
Let $S \subseteq \mathbb{S}$ and $|S| = s$. Let

$$Adv^S_{(i,j)}(h,k) = \begin{cases} 1 & \text{if the adversary can compute the} \\ & \text{common key between supernode} \\ & S_{i,j} \text{and} S_{h,k} \text{using the information} \\ & \text{stored in supernode } S_{m,n},\ (m,n) \in S \\ 0 & \text{elsewhere} \end{cases}$$

Let us denote,

$$P(S) = \frac{\sum_{(h,k)\in\mathbb{S}\setminus S} \sum_{\substack{(i,j)\in\mathbb{S}\setminus S \\ (i,j)\neq(h,k)}} (\mathsf{K}_{(i,j)}(h,k) - Adv^S_{(i,j)}(h,k))}{\sum_{(i,j)\in\mathbb{S}\setminus S} T(i,j)}$$

Then,

$$E'(s) = EXP(P(S)),$$

where EXP is the expectation over all $S \subseteq \mathbb{S}$ of size $|S| = s$.

We compare the experimental values of $E'(s)$ of our scheme with the experimental values of the key predistribution scheme for grid-group deployment by Ruj and Roy in [18]. Ruj and Roy considered similar deployment of sensor nodes as we did except that they used three supernodes per region whereas we used a single one. The supernodes are meant to provide interregion connectivity similar to our scheme. Both the schemes offer full connectivity between regions through supernodes. Ruj and Roy used transversal designs for key predistribution in supernodes. Figure 3 shows the comparison of the performance of our scheme with the scheme by Ruj and Roy in terms of $E'(s)$. The parameters of this graph can be found in Table 4. We considered a 37×37 square grid as the deployment zone in both the cases. In our scheme every square region contains one supernode and in Ruj and Roy scheme the number of supernodes per region is 3. Hence, the total number of supernodes is 1369 in our scheme and 4107 in Ruj-Roy scheme. The value of the security parameter of our key predistribution scheme is taken to be 4. We used C program to evaluate the values of $E'(s)$ for different values of s for both schemes. We compiled the source using GNU C compiler GCC 4.5.4. Figure 3 shows that our scheme is better than the scheme in [18] in terms of the number of interlinks broken when same number of supernodes are compromised in the hand of the adversary. So, for our

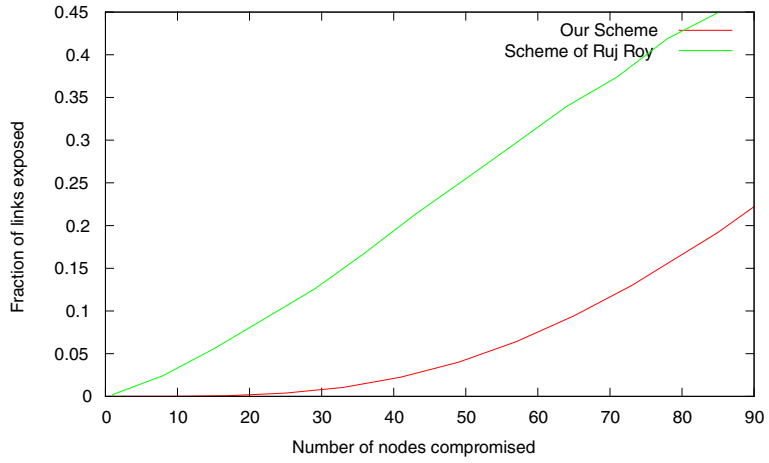

Figure 3 Graphical comparison of fraction of interlinks disconnected. This comparison is done with respect to the number of supernodes compromised for our scheme and the scheme in [18].

scheme, less number of links will get broken than the Ruj-Roy scheme when the same number of nodes are captured. So, in our scheme, more interregion links remain intact than the Ruj-Roy scheme when some supernodes are captured. Thus, our scheme exhibits better performance than the Ruj-Roy scheme though it makes use of only one-third of the number of supernodes used in Ruj-Roy scheme. Our scheme reduces the cost incurred due to the deployment of large number of supernodes and also enhances the resiliency of the network against node capture.

Definition 11. *$V'(s)$ is the fraction of groups that are disconnected from the rest of the groups with respect to the total number of groups when s number of supernodes are captured. In other words $V'(s)$ is the ratio of the number of groups that do not have any link to other groups after the s number of supernodes are captured to the total number of active supernodes present in the network before s many supernodes are captured.*

The result proved in Theorem 9 is also applicable for the interlinks between supernodes in different groups. Hence, for our scheme, individual groups do not get disconnected from the rest of the network unless a large number of supernodes get captured.

Table 4 Parameters used in comparison of the proposed scheme and the Ruj and Roy scheme in Figure 3

Parameters	Ruj-Roy scheme	Scheme of the current study
Number of square regions	1,369	1,369
Security parameter	-	4
Number of keys per node	13	-
Total number of nodes	4,107	1,369

Figure 4 shows the comparative performance of our scheme, and the Ruj-Roy scheme where the comparison is done in terms of $V'(s)$. The parameters of the graphical plot of Figure 4 is shown in Table 5. As defined above, $V'(s)$ is the fraction of nodes that get entirely disconnected from the rest of the network when s number of nodes get exposed. We used a 37×37 square grid in each case. The total number of supernodes in the entire network is 4, 107 in Ruj-Roy scheme and 1, 369 in our scheme. We have taken the security parameter of our scheme to be 4. The value of p in our scheme is 37. The number of keys (k) in a supernode is 23 in Ruj-Roy scheme. We used C program to evaluate the values of $E'(s)$ for different values of s for both schemes. We compiled the source using GNU C compiler GCC 4.5.4. Figure 4 shows that in our scheme, less number of nodes get detached from the network than the Ruj-Roy scheme in [18] when same number of nodes get captured by the adversary. Hence, our scheme is better than the Ruj-Roy scheme as it can keep more nodes connected to the network.

5.3 Overall resiliency

We shall now study the resiliency of the entire network taking into account all the groups, nodes, and supernodes.

We define $E''(s)$ as a new measure of overall resiliency in the entire network. It is defined to be the weighted average of the fractions of links exposed in every region $(i,j), 0 \leq i,j \leq N-1$ as well as the fraction of links exposed among the pair of supernodes when some nodes are compromised by the adversary in the entire network. The weight corresponding to the fraction of exposed links in a region (i,j) is equal to the number of pairs of uncompromised nodes present in that region (i,j). The weight corresponding to the fraction of exposed links between the supernodes are equal to the number of pairs of uncompromised

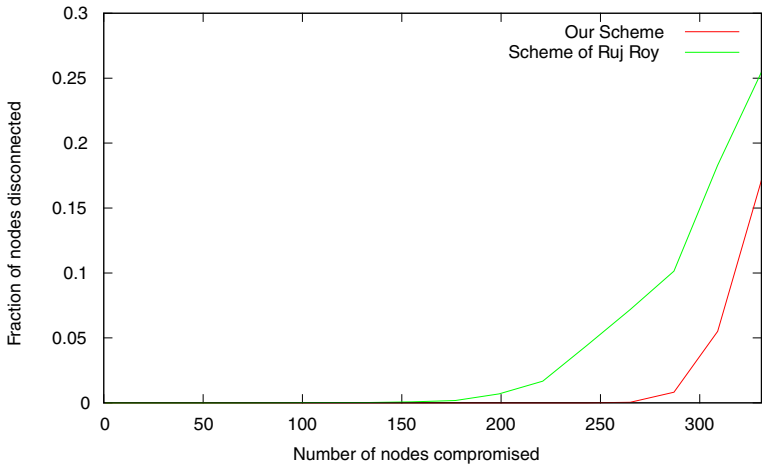

Figure 4 Graphical comparison of fraction of nodes disconnected. This comparison is done with respect to the number of nodes compromised for our scheme and the scheme in [18].

supernodes remaining in the network. We are the first to propose this as a measure of overall resiliency in terms of fraction of links exposed in the entire network. In this measure, we separately compute the values of fraction of links exposed($E(s_{ij})$) in every region $(i,j) : 0 \leq i,j \leq N-1$. We also measure the value of $E(s)$ among the set of supernodes in the network. Then, we compute the weighted average of all these values of $E(s)$.

Here, we take into account the entire network consisting of all the nodes and supernodes in all the regions. Let s_{ij} be the number of nodes compromised in group (i,j) and $s = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} s_{ij}$. Also, let s_g be the number of supernodes compromised. Hence, $0 \leq s_g \leq N^2$.

Let $E(s_{ij})$ be the value of fraction of links exposed in group (i,j) when s_{ij} many nodes are captured in group (i,j). Also, let $E^g(s_g)$ be the fraction of links exposed when s_g many supernodes are compromised. After s_{ij} many nodes are compromised in region (i,j), the number of uncompromised nodes present in region (i,j) is $\mathcal{N} - s_{ij}$. Hence, the weight corresponding to any region (i,j) is $\binom{\mathcal{N}-s_{ij}}{2}$ which is equal to the number of pairs of uncompromised nodes in region (i,j). Similarly, for the set of supernodes, the weight assigned is $\binom{N^2-s_g}{2}$. Therefore,

$$E''(s) = \frac{(\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{\mathcal{N}-s_{ij}}{2}E(s_{ij})) + \binom{N^2-g}{2}E^g(s_g)}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{\mathcal{N}-s_{ij}}{2} + \binom{N^2-g}{2}}. \quad (1)$$

Table 5 Parameters used in comparison of the proposed scheme and the Ruj and Roy scheme in Figure 4

Parameters	Ruj-Roy scheme	Scheme of the current study
Number of square regions	1,369	1,369
Security parameter	-	5
Number of keys per node	23	-
Total number of nodes	4,107	1,369

Hence, when the number of nodes captured from different groups is fixed, the overall $E''(s)$ is the weighted average of the value of $E(s_{ij})$ of all groups and the group of all supernodes.

Lemma 12. *When s_{ij} number of nodes are compromised in group $(i,j), 0 \leq i,j \leq N-1$ then $E''(s) < \max_{0\leq i,j<N}(E(s_{ij}))$ with a high probability where $s = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} s_{ij}$ and s is not-so-large.*

Proof. $E''(s) = \frac{(\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{\mathcal{N}-s_{ij}}{2}E(s_{ij}))+\binom{N^2-g}{2}E^g(s_g)}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{\mathcal{N}-s_{ij}}{2}+\binom{N^2-g}{2}}$

Hence,

$$E''(s) < \frac{(\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{\mathcal{N}-s_{ij}}{2}E(s_{ij})) + \binom{N^2-g}{2}E^g(s_g)}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{\mathcal{N}-s_{ij}}{2}}$$

Now, there are p^2+p+1 many nodes in any group which includes one supernode. If s_{ij} number of nodes are captured in group (i,j), the probability that the supernode will get captured is $\frac{s_{ij}}{p^2+p+1}$. In order to expose at least one link between two uncompromised supernodes, the adversary will have to compromise at least c nodes containing informations from the same key space of our scheme. The probability of compromising c many supernodes containing information from the same key space is very close to zero. Hence, $E^g(s_g) = 0$ with a high probability. So,

$$E''(s) < \frac{(\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{\mathcal{N}-s_{ij}}{2}E(s_{ij}))}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{\mathcal{N}-s_{ij}}{2}}$$

with a high probability, and the result follows from this. □

Corollary 13. *When s_{ij} number of nodes are compromised in group $(i,j), 0 \le i,j \le N-1$ then $E''(s) < \frac{c}{p^2+p+1}$ with a high probability where $s = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} s_{ij}$ and s is not-so-large and for all $(i,j) : 0 \le i,j < N, s_{ij} \le \frac{1}{2}c(c+1)$.*

Proof. Follows directly from *Lemma* 12 and *Theorem* 6. □

Corollary 13 gives an upper bound of the numeric value of fraction of links disconnected in the set of all uncompromised nodes of the network.

Definition 14. *$V''(s)$ is defined to be the weighted average of the fractions of nodes disconnected from the rest of the network in a region (i,j) or in the set of supernodes when some nodes get compromised. Here, the weights are proportional to the number of pairs of uncompromised nodes present among the nodes in any region or among the supernodes. We propose and apply this measure for the first time for measuring the resiliency for such deployment of wireless sensor network.*

Let $V(s_{ij})$ be the value of the fraction of nodes disconnected in region (i,j) when s_{ij} many nodes are captured. Again, let $s = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} s_{ij}$. Also let s_g be the number of supernodes captured by the adversary and $V^g(s_g)$ be the fraction of supernodes disconnected from other supernodes when s_g many supernodes are captured. After s_{ij} many nodes are compromised in region (i,j), the number of uncompromised nodes present in region (i,j) is $\mathcal{N} - s_{ij}$. Hence, the weight corresponding to any region (i,j) is $\binom{\mathcal{N}-s_{ij}}{2}$ which is equal to the number of pairs of uncompromised nodes in region (i,j). Similarly, for the set of supernodes, the weight assigned is $\binom{N^2-s_g}{2}$. Therefore,

$$V''(s) = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{\mathcal{N}-s_{ij}}{2}V(s_{ij}) + \binom{N-s_g}{2}V^g(s_g)}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{\mathcal{N}-s_{ij}}{2} + \binom{N-s_g}{2}}.$$

Lemma 15. *When s_{ij} number of nodes are compromised in group $(i,j), 0 \le i,j \le N-1$ then $V''(s) < \max_{0\le i,j<N}(V(s_{ij}))$ with a high probability where $s = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} s_{ij}$ and s is not so large.*

Proof. The proof is same as Lemma 12. □

Corollary 16. *When s_{ij} number of nodes are compromised in group $(i,j), 0 \le i,j \le N-1$ then $V''(s) = 0$ with a high probability where $s = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} s_{ij}$ and s is not-so-large and for all $(i,j) : 0 \le i,j < N, s_{ij} \le (p+1)c$.*

Proof. Follows immediately from Lemma 15 and Theorem 9. □

Corollary 16 provides a bound for the value of fraction of uncompromised nodes that get totally disconnected from the network.

We have done simulation of the performance of the key predistribution scheme for grid-group deployment taking $E''(s)$ and $V''(s)$ as the measure of the performance in the entire network. In this simulation, we randomly chose/compromised s many nodes from the entire network and then computed the values of $E''(s)$ and $V''(s)$ for them. Hence, it is equally probable for every chosen node to belong to a certain region. We measured the values of $E''(s)/V''(s)$ for any value of s by repeating the process 100 times and taking averages of the calculated values of the $E''(s)/V''(s)$ for this 100 iterations.

The value of $E''(s)$ for different values of s can be found in Table 6.

The values of $E''(s)$ for different values of the system parameters are obtained through simulation of the key predistribution model using C program. The first column of Table 6 shows the dimension of the grid used as deployment zone. The second column gives the number of nodes contained in a single group. The third column

Table 6 Values of $E''(s)$ for different values of s, size of grid and number of nodes in each group

Size of grid	Number of nodes in each group	Number of supernodes	Security parameter	s	Value of $E''(s)$
13	553	169	4	4,801	0.033041
14	307	196	4	3,001	0.010839
15	183	225	4	4,001	0.042171
18	183	324	4	5,001	0.025552
11	553	121	4	3,001	0.021120
15	381	225	3	3,126	0.034396
18	307	324	3	3,886	0.031764
16	381	256	3	5,626	0.105274
18	307	324	3	6,106	0.095601

shows the number of supernodes in the entire network which is equal to $R \times R$, R being the dimension of the square grid. The fourth column corresponds to the security parameter c. The fifth column gives the number of nodes compromised. The last column shows the values of $E''(s)$. It can be seen in Table 6 that as the grid size increases, the value of $E''(s)$ decreases while other parameters remain the same. So, the adversary needs to capture more nodes to damage the communication model considerably if the grid size is high enough. This happens as when the grid size increases, the total number of nodes in the network increases and the number of links between nodes also increases. It can be noted in this table that if the value of the security parameter is kept as low as 3 or 4, the key predistribution model can offer sufficient resiliency against node capture.

Table 7 gives the values of $V''(s)$ for different values of the number of captured nodes. It can be seen from Table 7 that the value of $V''(s)$ is very low even if a high number of nodes are captured. So, the key predistribution model is highly resilient as far as the $V''(s)$ is concerned. Also, if the size of the grid is increased, the value of $V''(s)$ gets reduced.

5.4 Comparison with other schemes

Next, we compare our proposed scheme with some other key predistribution schemes that use deployment knowledge. These schemes include Du et al. 2004 [20] and 2006 [21], Liu and Ning 2003 [39] and 2005 [40], Yu and Guan 2005 [28] and 2008 [29], Zhou et al. 2006 [23], Huang et al. [24], Huang and Medhi 2007 [25], Chan and Perrig 2005 [26], Simonova et al. 2006 [27].

Huang et al. [24,25] used rectangular deployment zone which is divided into equal-sized regions of smaller size. In this scheme, the sensors randomly choose the keys. Huang et al. used multispace Blom scheme [4] for key predistribution. In this scheme, all nodes are identical with respect to the amount of resources they possess. This is where this scheme is different from ours. In our scheme, there are two different types of nodes viz. common nodes and agents giving rise to a heterogeneous network. Moreover, in Huang et al. scheme, the nodes in a region can communicate directly with each other with probability of >0.5; whereas, in our scheme, they can do so with a probability equal to 1 as our scheme ensures full interregion connectivity. Hence, in this scheme, more amount of computation will be required for communication than our scheme. The scheme of Huang et al. is perfectly secure against selective and random node capture attack. Hence, capture of some number of nodes by an adversary will have negligible effect to the links among the uncompromised nodes. However, if we take all the links of compromised and uncompromised nodes into account, then the fraction of links compromised will be higher.

Zhou et al. [23] used two types of sensor nodes viz. static and mobile. This scheme uses pairwise keys with each sensor within the same region. Hence, it requires high amount of memory to hold the pairwise keys if the number of sensors within a region is high enough. If there are n number of nodes within a region, then the number of keys to be stored in a node is $O(n^2)$ under the Zhou et al. scheme; whereas, it is $O(\sqrt{n})$ in Çamptepe and Yener scheme which is used in our key predistribution scheme. Hence, our scheme is much better than Zhou et al. in terms of memory efficiency.

Liu and Ning [39,40] used deployment knowledge. There, the whole deployment zone is split into smaller square regions like our scheme. However, in their schemes, only a single node is deployed in a square region as opposed to our scheme where there are a group of nodes deployed in a region. They used the polynomial-based scheme of Blundo et al. [5]. The deployment region is broken down into equal-sized squares $\{C_{i_c,i_r}\} i_c = 0, 1, \ldots, C-1, i_r = 0, 1, \ldots, R-1$,

Table 7 Values of $V''(s)$ for different values of s, size of grid and number of nodes in each group

Size of grid	Number of nodes in each group	Number of supernodes	Security parameter	s	Value of $V''(s)$
14	553	196	3	24,000	0.114369
15	307	225	3	20,000	0.187892
14	183	196	3	18,009	0.841926
11	381	121	4	24,000	0.959935
15	307	225	4	21,002	0.027675
13	871	169	3	25,000	0.033112
7	553	49	3	6,000	0.112729
9	553	81	4	11,000	0.030479
14	307	196	4	14,000	0.000322
7	381	49	3	10,000	0.976503

each of which is a cell with coordinates (i_c, i_r) denoting row i_r and column i_c . Each of the cells is associated with a bivariate polynomial. For a $R \times C$ grid, the setup server generates RC t-degree polynomials $\{f_{i_c,i_r}(x,y)\} i_c = 0,1,\ldots,C-1, i_r = 0,1,\ldots,R-1$, and assigns $f_{i_c,i_r}(x,y)$ to cell C_{i_c,i_r}. For each sensor, the setup server determine its home cell and its four neighboring cells which lie adjacent to the home cell in the same row and column. The setup server distributes to the sensor the coordinates of the home cell and the polynomial shares of the home cell and its neighboring cell. For example, for a sensor U_u in the cell with coordinate (r', c'), the polynomial shares $f_{r'-1,c'}(u,y), f_{r',c'-1}(u,y), f_{r'+1,c'}(u,y), f_{r',c'+1}(u,y), f_{r,c}(u,y)$ are given. For direct key establishment, a node broadcasts the coordinates of its home cell. From this coordinate, the destination node finds out the common polynomial that it shares with the broadcasting node if at all. Now, the common key can be calculated using the same method as [5].

In Simonova et al.'s [27] scheme, the number of specialized nodes depends upon the size of the network unlike ours which is constant (=1). The resiliency as given in the graph is much lower compared to our scheme. Also, resiliency in terms of nodes or regions disconnected has not been presented.

Du et al. [21] proposed another key predistribution using deployment knowledge that uses multiple space Blom scheme [4]. Under this scheme, sensors randomly choose keys from a set of different instances of Blom space. Unlike our scheme, this scheme does not guaranty full connectivity.

As we have discussed earlier, the key predistribution scheme of Ruj and Roy in [18] uses deployment knowledge. Similar to our scheme, this scheme uses the Çamptepe and Yener scheme for key predistribution within the same region. This scheme exhibits lower resiliency among the set of agents that provide interregion connectivity as discussed in previous sections. In other words, our scheme offers more resilient interregion connectivity than Ruj and Roy scheme.

Figure 5 shows a pictorial comparison of our scheme with standard schemes that use deployment knowledge. This comparison is based on the values of fraction of total links broken when some nodes get captured. This comparison takes into account all the links in the network which includes the links in compromised nodes as well. The parameters of the different schemes are following:

DDHV scheme has parameters $k = 200, \omega = 11$, and $\tau = 2$. LN scheme has parameters $k = 200, m = 60$, and $L = 1$; YG scheme has parameters $k = 100$; ZNR scheme has parameters $k = 100$; HMMH scheme has parameters $k = 200, \omega = 27$, and $\tau = 3$; SLW scheme has parameters $k = 16, p = 11$, and $m = 4$; Ruj-Roy scheme has parameters $k = 12$. Our scheme has parameters $p = 11$ and $c = 4$. The size of the network in DDHV, LN, YG, ZNR, and HMMH is 10,000; for SLW, it is 12,100. It is 16,093 for Ruj-Roy scheme and in our scheme. We simulated the behavior of the key predistribution schemes for random node capture attack. All schemes are implemented identical network. It can be seen in Figure 5 that our scheme offers better performance than similar schemes that make use of deployment knowledge up to a certain limit of the number of nodes captured by the adversary. We used C program for running the simulation.

The reason why our scheme excels in performance can be inferred from Lemma 3 and Proposition 4. Lemma 3 says that in order to compromise the links between any two nodes, the adversary is required to compromise at least c (c is the security parameter) nodes having

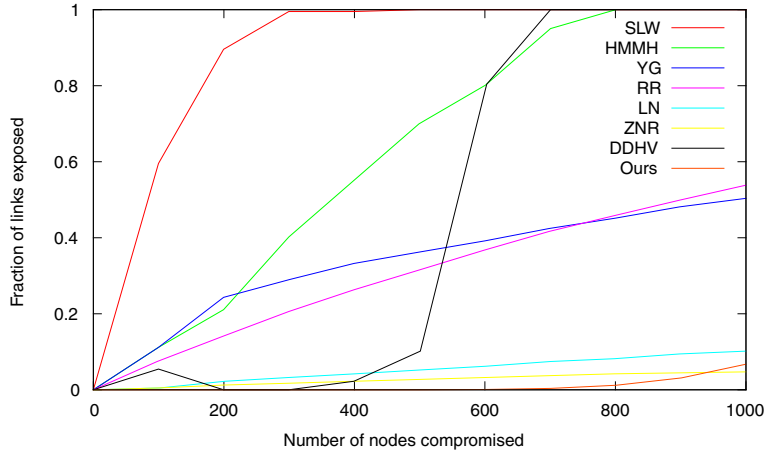

Figure 5 Graphical comparison of fraction of links disconnected. This comparison is done with respect to the number of nodes compromised for our scheme and the schemes in [18,20,21,23-29,39,40].

information from the same key space as the two nodes. However, in most of the other schemes, the same thing can be done by capturing a single node. Again, Proposition 4 says that the probability of existence of a link between a pair of nodes is high even if many nodes are compromised. So, even if the number of captured nodes is high enough, the value of fraction of broken links can be very low in our scheme. This fact is corroborated by the performance of our scheme as shown in Figure 5.

We present a comparative study of communication, storage, and scalability of several schemes in Table 8. This table gives a comparison with respect to communication, storage cost, etc. of our scheme and the schemes in [18,21,23-29,39,40]. The first column of Table 8 shows the name of the scheme. The second column corresponds to the type of deployment used by the key predistribution scheme. The third column shows the type of nodes in the WSN. There are two types of sensor nodes viz. homogeneous and heterogeneous. All the nodes in a homogeneous network are identical in terms of the resources they possess. However, in heterogeneous networks, there are different types of nodes who mainly differ in the amount of computational resource built inside them. The fourth column shows the communication cost of each key predistribution scheme. When two nodes wish to communicate, they need to exchange some information before a secure communication can start. This information may be their unique identifiers or something else that is required to compute the shared key between them. The storage column gives the amount of memory needed to store the keys a node. Here, N is the number of sensors in the network, and g is the number of groups. The last column says whether the key predistribution scheme is scalable or not. The communication cost of our scheme is $O(\log N)$, and the storage overhead is $O(N^{\frac{1}{4}})$. Our scheme consumes less amount of memory than other schemes except the DDHV scheme in [20,21] and the Yu-Guan scheme in [28,29] that uses constant amount of storage. However, our scheme outperforms both of them in terms of resiliency measure used in the comparison in Figure 5.

6 Conclusions

In this paper, we have presented a key predistribution scheme for a wireless sensor for a grid-group-based deployment. Here, the entire deployment zone is a square which is divided into a number of smaller squares. Each square is identical in terms of physical area and number of sensor nodes. The sensor nodes belonging to a smaller square form a group among themselves. All the groups contain two types of nodes viz. ordinary and nodes. A node within a group can make direct communication to any other node in the same group or region. Nodes belonging to two different group communicate via special nodes called nodes. These nodes are more resourceful than ordinary nodes in terms of memory, computational power, and energy. We used two types of different key predistribution schemes for this deployment. The ordinary sensor nodes and the node within a group use symmetric design-based key predistribution scheme proposed in [6] for within group communication. The nodes contain two types of keys. It can communicate to other sensor nodes belonging to the same group. Moreover, it can communicate with other nodes by means of a separate key predistribution scheme. Our scheme offers better resiliency than other existing schemes like the most notable scheme by Ruj & Roy [18] and the Zhou et al. scheme in [23].

Table 8 Comparison of schemes with respect to type of deployment, node, communication, and storage overhead and scalability

Schemes	Deployment	Nodes	Communication cost	Storage	Scalability
DDHV [20,21]	Grid-group	Homogeneous	$O(1)$	$O(1)$	Scalable
LN [39,40]	Grid	Homogeneous	$O(\log N)$	$O(\sqrt{N})$	Not scalable
YG [28,29]	Grid-group	Homogeneous	$O(1)$	$O(1)$	Not scalable
ZNR [23]	Group	Heterogeneous	$O(\log N)$	$O(N/g)$[a]	Not scalable
				$O(N)$[b]	
HMMH [24]	Grid-group	Homogeneous	$O(1)$	$O(\sqrt{N})$	Scalable
HM [25]	Grid-group	Homogeneous	$O(1)$	$O(\sqrt{N})$	Scalable
PIKE [26]	Grid	Homogeneous	$O(\log N)$	$O(\sqrt{N})$	Not scalable
SLW [27]-2	Grid-group	Heterogeneous	$O(\log N)$	$O(\sqrt{N/g})$	Scalable
Ruj-Roy [18]	Grid-group	Heterogeneous	$O(\log N)$	$O(N^{\frac{1}{4}})$[a]	Not scalable
				$O(N^{\frac{1}{4}})$[b]	
Current scheme	Grid-group	Heterogeneous	$O(\log N)$	$O(N^{\frac{1}{4}})$[a]	Not scalable
				$O(N^{\frac{1}{4}})$[b]	

Here, N is the total number of sensors in the network. g is the number of groups in the network. [a]the storage for small sensor nodes, and [b] the storage for agents.

We have shown that our scheme ensures that there will be high probability of existence of a common unexposed link between two nodes belonging to two different groups even if a considerable number of nodes are compromised by the adversary.

Competing interests
The authors declare that they have no competing interests.

Received: 16 October 2012 Accepted: 16 May 2013
Published: 30 May 2013

References

1. L Eschenauer, VD Gligor, in *ACM Conference on Computer and Communications Security*, ed. by V Atluri. A key-management scheme for distributed sensor networks (ACM New York, 2002), pp. 41–47
2. H Chan, A Perrig, DX Song, in *IEEE Symposium on Security and Privacy*. Random key predistribution schemes for sensor networks (IEEE Computer Society, Berkeley, CA, USA, 11–14 May 2003), p. 197
3. D Liu, P Ning, in *ACM Conference on Computer and Communications Security*, ed. by S Jajodia, V Atluri, and T Jaeger. Establishing pairwise keys in distributed sensor networks (ACM New York, 2003), pp. 52–61
4. R Blom, in *Proceedings of EUROCRYPT 84, A Workshop on the Theory and Application of Cryptographic Techniques*, vol. 209, Paris, ed. by T Beth, N Cot, and I Ingemarsson. An optimal class of symmetric key generation systems (Springer Berlin, 1984), pp. 335–338
5. C Blundo, AD Santis, A Herzberg, S Kutten, U Vaccaro, M Yung, in *Advances in Cryptology–CRYPTO '92*, vol. 740, ed. by EF Brickell. Perfectly-secure key distribution for dynamic conferences (Springer Berlin, 1992), pp. 471–486
6. SA Çamtepe, B Yener, in *Computer Securtiy–ESORICS*, vol. 3193, ed. by P Samarati, PYA Ryan, D Gollmann, and R Molva. Combinatorial design of key distribution mechanisms for wireless sensor networks (Springer Berlin, 2004), pp. 293–308
7. SA Çamtepe, B Yener, Combinatorial design of key distribution mechanisms for wireless sensor networks. IEEE/ACM Trans. Netw. **15**(2), 346–358 (2007)
8. J Lee, DR Stinson, in *IEEE Wireless Communications and Networking Conference*. A combinatorial approach to key predistribution for distributed sensor networks (IEEE, New Orleans, 13–17 Mar 2005)
9. J Lee, DR Stinson, in *Selected Areas in Cryptography*, ed. by Handschuh, and Hasan. Deterministic key predistribution schemes for distributed sensor networks (Springer Berlin, 2004), pp. 294–307
10. S Ruj, BK Roy, in *ISPA, Parallel and Distributed Processing and Applications*, vol. 4742, ed. by I Stojmenovic, RK Thulasiram, LT Yang, W Jia, M Guo, and RF de Mello. Key predistribution using partially balanced designs in wireless sensor networks (Springer Berlin, 2007), pp. 431–445
11. D Chakrabarti, S Maitra, BK Roy, in *Information Security*, vol. 3650, ed. by J Zhou, J Lopez, RH Deng, and F Bao. A key pre-distribution scheme for wireless sensor networks: merging blocks in combinatorial design (Springer Berlin, 2005), pp. 89–103
12. S Bag, S Ruj, in *IEEE Workshops of International Conference on Advance Information Networking and Applications (WAINA)*. Key distribution in wireless sensor networks using finite affine plane (Singapore, 22–25 Mar 2011), pp. 436–441
13. CJ Mitchell, F Piper, Key storage in secure networks. Discrete Appl. Math. **21**(3), 215–228 (1988)
14. J Dong, D Pei, X Wang, in *Information Security and Cryptology*. A key predistribution scheme using 3-designs (Springer Berlin, 2007)
15. J Dong, D Pei, X Wang, A class of key predistribution schemes based on orthogonal arrays. JCST. **23**, 825–831 (2008)
16. S Blackburn, T Etzion, K Martin, M Paterson, in *Information Theoretic Security. Lecture Notes in Computer Science 5155*, ed. by S Fehr. Efficient key predistribution for grid-based wireless sensor networks (Springer Berlin, 2008), pp. 54–69
17. R Wei, J Wu, in *Selected Areas in Cryptography*, ed. by Handschuh, and Hasan. Product construction of key distribution schemes for sensor networks (Springer Berlin, 2004), pp. 280–293
18. S Ruj, BK Roy, Key predistribution using combinatorial designs for grid-group deployment scheme in wireless sensor networks. TOSN. **6**(1), 4:1–4:28 (2009)
19. S Ruj, BK Roy, in *Information Security and Cryptology*, vol. 5487, ed. by M Yung, P Liu, and D Lin. Key predistribution schemes using codes in wireless sensor networks (Springer Berlin, 2008), pp. 275–288
20. W Du, J Deng, YS Han, S Chen, PK Varshney, in *Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications*, vol. 1. A key management scheme for wireless sensor networks using deployment knowledge (IEEE, Hong Kong, 7–11 Mar 2004)
21. W Du, J Deng, YS Han, PK Varshney, A key predistribution scheme for sensor networks using deployment knowledge. IEEE Trans. Dependable Sec. Comput. **3**, 62–77 (2006)
22. D Liu, P Ning, Improving key predistribution with deployment knowledge in static sensor networks. TOSN. **1**(2), 204–239 (2005)
23. L Zhou, J Ni, Ravishankar C V, in *Proceedings of 25th IEEE International Conference on Computer Communications*. Supporting secure communication and data collection in mobile sensor networks (Barcelona, 23–29 April 2006)
24. D Huang, M Mehta, D Medhi, L Harn, in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*. Location-aware key management scheme for wireless sensor networks, Washington (ACM New York, 2004), pp. 29–42
25. D Huang, D Medhi, Secure pairwise key establishment in large-scale sensor networks: an area partitioning and multigroup key predistribution approach. TOSN. **3**(3) (2007)
26. H Chan, A Perrig, in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1. PIKE: peer intermediaries for key establishment in sensor networks (Miami, 13–17 Mar 2005), pp. 524–535
27. K Simonova, ACH Ling, XS Wang, in *Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, ed. by S Zhu, Liu D. Location-aware key predistribution scheme for wide area wireless sensor networks (ACM New York, 2006), pp. 157–168
28. Z Yu, Y Guan, in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks*. A key pre-distribution scheme using deployment knowledge for wireless sensor networks (IEEE, Los Angeles, 15 Apr 2005), pp. 261–268
29. Z Yu, Y Guan, A key management scheme using deployment knowledge for wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. **19**(10), 1411–1425 (2008)
30. S Bag, in *4th International Conference, CNSA 2011*. Key predistribution in 3-dimensional grid-group deployment scheme (Springer, Chennai, India, 15–17 Jul 2011), pp. 302–319
31. DR Stinson, *Combinatorial Designs: Construction and Analysis*. (Springer, New York, 2004)
32. AP Street, DJ Street, *Combinatorics of Experimental Design*. (Clarendon Press, Oxford, 1987)
33. DC Lay, *Linear Algebra and Its Applications*, 3rd edn. (Addison Wesley, 75 Arlington Street, Boston, MA 02116, 2005)
34. CD Meyer, *Matrix Analysis and Applied Linear Algebra*. (Society for Industrial and Applied Mathematics (SIAM) Philadelphia, United States, 2001)
35. W Du, J Deng, YS Han, PK Varshney, in *Intelligence and Security Informatics*, ed. by S Jajodia, V Atluri, and T Jaeger. A pairwise key pre-distribution scheme for wireless sensor networks (Springer Berlin, 2003), pp. 42–51
36. FJ MacWilliams, NJA Sloane, *The Theory of Error Correcting Codes*. (Northland Holland, Amsterdam, 1988)
37. RD Pietro, LV Mancini, A Mei, Energy efficient node-to-node authentication and communication confidentiality in wireless sensor networks. Wireless Netw. **12**(6), 709–721 (2006)
38. J Lee, DR Stinson, On the construction of practical key predistribution schemes for distributed sensor networks usin combinatorial designs. ACM Trans. Inf. Syst. Secur. **11**(2) (2008)
39. D Liu, P Ning, in *Proceedings of the 10th ACM conference on Computer and communications security*. Establishing pairwise keys in distributed sensor networks (ACM New York, 2003), pp. 52–61
40. D Liu, P Ning, Improving key predistribution with deployment knowledge in static sensor networks. TOSN. **1**(2), 204–239 (2005)

doi:10.1186/1687-1499-2013-145
Cite this article as: Bag and Roy: **A new key predistribution scheme for general and grid-group deployment of wireless sensor networks.** *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:145.