Key Predistribution Using Combinatorial Designs for Grid-Group Deployment Scheme in Wireless Sensor Networks

SUSHMITA RUJ and BIMAL ROY
Indian Statistical Institute

We propose a new grid-group deployment scheme in wireless sensor networks. We use combinatorial designs for key predistribution in sensor nodes. The deployment region is divided into square regions. The predistribution scheme has the advantage that all nodes within a particular region can communicate with each other directly and nodes which lie in a different regions can communicate via special nodes called agents which have more resources than the general nodes. The number of agents in a region is always three, whatever the size of the network. We give measures of resiliency taking the Lee distance into account. Apart from considering the resiliency in terms of fraction of links broken, we also consider the resiliency as the number of nodes and regions disconnected when some sensor are compromised. This second measure, though very important, had not been studied so far in key predistribution schemes which use deployment knowledge. We find that the resiliency as the fraction of links compromised is better than existing schemes. The number of keys preloaded in each sensor node is much less than all existing schemes and nodes are either directly connected or connected via two hop paths. The deterministic key predistribution schemes result in constant-time computation overhead for shared key discovery and path key establishment.

Categories and Subject Descriptors: C.2 [**Computer-Communication Networks**]: General—*Security and protection*

General Terms: Security, Design, Algorithms

Additional Key Words and Phrases: Combinatorial design, deployment, key predistribution, Lee distance, secure communication, transversal design

ACM Reference Format:

Ruj, S. and Roy, B. 2009. Key predistribution using combinatorial designs for grid-group deployment scheme in wireless sensor networks. ACM Trans. Sensor Netw. 6, 1, Article 4 (December 2009), 28 pages.

1. INTRODUCTION

Sensor nodes in adversarial areas are prone to node capture or node compromise. To enhance the resiliency against node compromise, sensor nodes may be

Authors' address: S. Ruj and B. Roy, Applied Statistics Unit, Indian Statistical Institute, 203 Barrackpore Trunk Road, Kolkata 700108, India; email: {sush_r,bimal}@isical.ac.in.

deployed in groups in a predetermined way. Several studies have been made where deployment knowledge has been used instead of deploying nodes randomly. Schemes in which deployment knowledge has been used are Du et al. [2004, 2006], Yu and Guan [2005, 2008], Liu and Ning [2003, 2006], Younis et al. [2006], Zhou et al. [2006], Huang et al. [2004], Huang and Medhi [2007], Chan and Perrig [2005], and Simonova et al. [2006].

Towards secure communication, it is important that any two sensor nodes should communicate in an encrypted manner using a common secret key. Due to resource constraints, symmetric encryption is preferred over public key techniques. Hence key predistribution is an effective solution. Given any kind of deployment, the key predistribution techniques may be randomized, deterministic, or hybrid. Surveys of key predistribution schemes can be found in Ruj [2009], Çamtepe and Yener [2005], and Xiao et al. [2007].

In a homogeneous key predistribution scheme all sensor nodes have the same storage and computation power. In this article we propose a heterogeneous key predistribution scheme for a known deployment scheme. Other heterogeneous schemes can be found in Zhu et al. [2003], Jolly et al. [2003], Cheng and Agrawal [2007], Paterson and Stinson [2008], Das and Sengupta [2008], Heinzelman et al. [2000], Ferreira et al. [2005], Oliveira et al. [2006, 2007], Du et al. [2007], and Hussain et al. [2007]. In some applications where sensors are scattered over an adversarial area we require that the complete disconnection of one region does not affect another region. For example, consider the situation where sensors are deployed in a battlefield. Suppose the adversary captures one region and captures all the sensors. We must ensure that other regions are not affected by such a compromise. For this the whole target region where the sensor nodes are to be deployed is partitioned into equal-sized squares or grids as in Liu and Ning [2003, 2005]. There are two types of sensor nodes having different power and storage capacities: the nodes that have lower storage and battery power and agents which are more powerful. It is also assumed that it is more difficult to compromise an agent than a sensor node. The 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 keys are preloaded in such a way that all the nodes belonging to a particular square region can communicate with each other directly.

Key predistribution for these nodes is done using combinatorial designs called *projective planes*. The key predistribution scheme is given in Section 3. It is to be noted that any other combinatorial design could be used like PBIBD, as discussed in Ruj and Roy [2007]. Each of the square regions also has some specialized nodes called *agents* with higher battery power and communication range. Since the regions can be thought of as a grid, we consider the transmission range of each agent (in a region of the grid) to be a *Lee sphere* of appropriate radius [Blackburn et al. 2008]. We call this radius as the *Lee distance*. Any two agents cannot communicate with each other outside the Lee sphere. So all resiliency calculations have been done taking the Lee distance into account. Only three agents are required (whatever be the size of the network)

in each region to ensure that a particular region can communicate with all other regions within communication range. Agents have two types of keys. One type of keys is used for communicating within the square region and the other type of keys is used for communicating with agents in different regions. The key predistribution scheme used for interregion communication makes use of transversal designs. In any key predistribution schemes, three steps are followed: key predistribution, shared key discovery (finding one or more common keys which can be used for communicating securely), and path-key establishment (which is invoked when nodes do not share a common key). The reason for using combinatorial designs is that they have "patterns" which make shared key discovery and path-key establishment very efficient both in terms of computation and communication complexity. This very important observation was pointed out by Lee and Stinson [2008] and Ruj and Roy [2008].

We discuss two types of attacks: selective node capture attack and random node capture attack. We measure the resiliency in terms of two parameters. We give experimental values of resiliency and analyze how nodes and agents will be affected when sensor nodes are compromised. We measure the resiliency in terms of the fraction of links compromised and also the fraction of nodes and regions disconnected. This second parameter has not been so far discussed in any key predistribution scheme using deployment knowledge. We obtain very high resiliency (in terms of fraction of links compromised) compared to the schemes in Du et al. [2004, 2006], Yu and Guan [2005, 2008] Liu and Ning [2003, 2005], Younis et al. [2006], Zhou et al. [2006], Huang et al. [2004], Huang and Medhi [2007], Chan and Perrig [2005], and Simonova et al. [2006]. We also get a very good resiliency in terms of the fraction of nodes disconnected and regions disconnected. For example, for a 31×31 grid, if nodes in each region contain 18 keys ($p = 17$), then on an average 17298 nodes must be compromised to disconnect one node of each region. We note that for a 31×31 grid if 200 agents are compromised then about $2 - 3$ regions will be disconnected, where each agent contains 20 keys. If agents contain 25 keys then on compromising 200 agents only about one region will be disconnected.

We use combinatorial structures like projective planes and transversal designs. The deterministic designs help in the estimation of resiliency. Direct communication is assured between any two nodes in the same region.

The rest of this article is organized in the following way. In Section 2, we define basic concepts. In Section 3, we present a grid-group deployment scheme and predistribution scheme. In Section 4, we study the resiliency of the network. We give two parameters for security. In Section 5, we make a comparison of our scheme with other schemes. The conclusion and future work are presented in Section 6.

2. PRELIMINARIES

A *set system* or *design* [Lee and Stinson 2005] is a pair (X, A), where A is a set of subsets of X, called *blocks*. The elements of X are called varieties. A

Table I. Mapping from Set System to Sensor Networks

Blocks	sensor nodes
Elements	Key identifiers
Elements present in each block	Keys present in each sensor

$BIBD(v, b, r, k, \lambda)$, is a *design* which satisfies the following conditions:

(1) $|X| = v$, $|A| = b$,
(2) each subset in A contains exactly k elements,
(3) each variety in X occurs in r blocks,
(4) each pair of varieties in X is contained in exactly λ blocks in A.

When $v = b$, the *BIBD* is called a *symmetric BIBD* and denoted by $SBIBD(v, k; \lambda)$.

A *finite projective plane* consists of a finite set of points P and a set of subset of P called *lines*. For an prime power q a finite projective plane consists of q^2+q+1 points, q^2+q+1 lines where each line contains $q+1$ points and each point occurs in $q+1$ lines. If we consider lines as blocks and points as elements, then a *finite projective plane of order* q is a $SBIBD(q^2 + q + 1, q + 1; 1)$.

A *transversal design* [Street and Street 1987, Section 6.3] $TD(k, \lambda; r)$, with k groups of size r and index λ, is a triple (X, G, A) where:

(1) X is a set of kr many elements (varieties),
(2) $G = \{G_1, G_2, \cdots, G_k\}$ is a family of k sets (each of size r) which form a partition of X,
(3) A is a family of k-sets (or blocks) of varieties such that each k-set in A intersects each group G_i in precisely one variety, and any pair of varieties which belong to different groups occur together in precisely λ blocks in A.

We denote a transversal design with $\lambda = 1$ as $TD(k, r)$. It can be shown that if there exists a $TD(k, r)$, then there exists a (v, b, r, k) design with $v = kr$, $b = r^2$. We present a construction of a transversal design $TD(k, r)$ [Lee and Stinson 2005].

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

We define a block $A_{i,j}$ by

$$A_{i,j} = \{(x, xi + j \bmod r) : 0 \leq x < k, \ r \text{ is prime}\}. \tag{1}$$

The mapping of set systems to sensor networks has been presented in Table I.

The *Manhattan distance* between two points is the sum of the horizontal and vertical distance between the points.

Consider a square grid (as shown in Figure 1). A *Lee sphere* [Blackburn et al. 2008] of radius ρ centered at a given square consists of the set of the

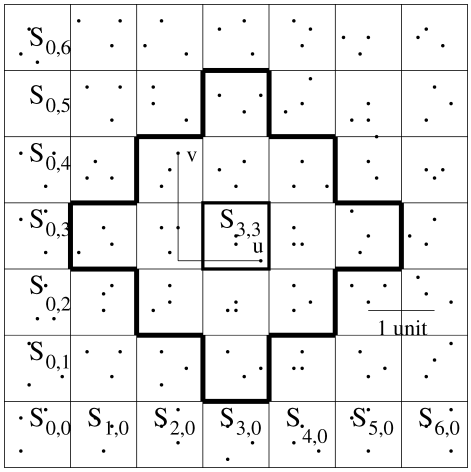


Fig. 1. A deployment region having 49 regions. Lee distance is 2. Each region has three agents. Region $S_{3,3}$ is connected to all the regions within the marked Lee sphere. Though u and v are not within Manhattan distance 2, but for simplicity v is considered to be within Lee distance 2 from u.

Table II. Notations

r	Dimension of the grid, r is a prime
N	Number of agents in the network
$S_{i,j}$	(i, j)th region
$P_{i,j}$	Set of all keys assigned to nodes in the $S_{i,j}$th region
$p+1$	Number of keys in each node
p^2+p+1	Maximum number of sensor nodes in each region
k	Number of Type I, Type II or Type III keys present in each agent
$B^1_{i,j}$	Set of keys assigned to agent of Type I in the region $S_{i,j}$
$B^2_{i,j}$	Set of keys assigned to agent of Type II in the region $S_{i,j}$
$B^3_{i,j}$	Set of keys assigned to agent of Type III in the region $S_{i,j}$
ρ	Lee distance
$L(x, y)$	Interlinks connected by key (x, y), $x, y < r$
$E'(s)$	Number of intralinks broken when s nodes are compromised
$E''(s)$	Number of interlinks broken when s agents are compromised
$V'(s)$	Number of nodes disconnected when s nodes are compromised
$V''(s)$	Number of agents disconnected when s agents are compromised

squares that lie at a distance of at most ρ from the square. ρ is called the Lee distance.

Table II represents the notations we have used throughout the article.

2.1 Threat Model

There are different types of models for node capture [Pietro et al. 2006].

(1) *Random node capture attack.* Nodes are captured randomly.

(2) *Selective node capture attack.* This capture attack is given in Pietro et al. [2006]. Assume that the attacker's goal is to collect a subset T of the keys

in the pool. The attacker has already compromised a number of sensors, and has collected all their keys in a set W. For every sensor s in the WSN, the *key information gain* $G(s)$ is a random variable equal to the number of keys in the key ring of s which are in T and are not in W. For example, if the attacker's goal is to compromise the channel between sensors s_a and s_b, subset T in the preceding definition is equal to $M_a \bigcap M_b$, (where M_a and M_b are the keys in the key chains of s_a and s_b, respectively) that is, it contains all the keys which are in the key ring of both s_a and s_b . Assuming that the attacker has collected a set W of keys, random variable $G(s)$ is equal to $|(M_s \bigcap M_a \bigcap M_b) \backslash W|$. 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 key ring.

In this article we show that an attacker does not gain in any way by launching a selective node capture attack. In fact selective node capture is just as good as random node capture from the point of view of the attacker.

3. KEY PREDISTRIBUTION SCHEME

We first discuss the deployment architecture and then present our key predistribution scheme. When nodes are deployed in a region, all nodes need not communicate with all other nodes in the network. Due to limited power, all nodes cannot communicate with all other nodes. So we divide the entire region into equal-sized squares or grids as done in Liu and Ning [2003, 2005], and Huang and Medhi [2007]. Let us consider an $r \times r$ deployment area, consisting of r^2 regions (r is a prime). The r^2 regions are numbered as $S_{i,j}$, $0 \leq i, j < r$ as shown in Figure 1. All nodes within a particular square region communicate with each other. However, to communicate with nodes in other regions, there are specialized nodes called *agents*. These are more powerful than the nodes and have higher storage capacities. Let each agent have a transmission range of ρ. Then each agent can communicate directly with agents which lie within the circle of radius ρ and center as the agent itself and share keys with the agent. When we consider communication between regions we consider the communication region as the Lee sphere (defined in Section 2) of appropriate radius [Blackburn et al. 2008]. Though the Lee sphere does not exactly depict the agents within Manhattan distance from a particular agent, for simplicity we consider the Lee spheres around an agent in all our calculations. Each region contains a set of three agents, irrespective of the number of regions. In general the number of agents must be proportional to the number of regions. However, three and only three agents are enough to ensure that a region can communicate with all regions within Lee distance. This is formalized in Theorem 3.1. Throughout the article we refer to any small node or an agent as a *sensor node*.

3.1 Key Predistribution in Nodes in a Region

For each region keys are predistributed in the nodes independently of the other region using some existing predistribution scheme. There are many randomized

[Eschenauer and Gligor 2002; Liu and Ning 2003, 2005; Chan et al. 2003; Hwang and Kim 2004], deterministic [Lee and Stinson 2005, 2004; Çamtepe et al. 2006; Çamtepe and Yener 2004; Ruj and Roy 2007; Ruj et al. 2008], and hybrid [Çamtepe and Yener 2004; Chakrabarti et al. 2006] schemes of predistribution available in the literature. We use deterministic design because we would like to ensure that all nodes within a region can communicate with each other directly. Probabilistic designs cannot guarantee this fact. Since deterministic designs have a pattern, shared-key discovery and path-key establishment is efficient [Lee and Stinson 2008]. We choose the symmetric design as given in Çamtepe and Yener [2004, 2007]. Any other combinatorial design which ensures direct communication can be employed like that given in Ruj and Roy [2007]. Each of the smaller regions consists of $p^2 + p - 2$ nodes each containing $p + 1$ keys, where p is a prime power. We do not use the other designs using generalized quadrangles given in Çamtepe and Yener [2004]. Though these designs result in large network size (of the order of 3 in the number of keys), the connectivity is very low. These designs also have a large size of key pool (also of the order of 3 in the number of keys), however, result in very low connectivity. A low connectivity results in the higher computation for path-key establishment and hence may deplete the battery power quickly. A large key pool size has the advantage that the resiliency is high. So there is a trade-off between resiliency and connectivity. Hence depending upon the application we can choose the key predistribution schemes. Here we choose the symmetric design. In case the number of sensors is not of the form $p^2 + p + 1$ for some prime power p, then we choose p such that $n \leq p^2 + p + 1$ and distribute keys to only n sensors. So first n is decided upon. Based on the value of n the parameter p is chosen. A maximum of $r^2(p^2 + p + 1)$ sensor nodes (nodes and agents) can be supported. If $r = 23$, $p = 17$, then 162403 sensor nodes can be supported. We distribute the keys according to Algorithm 1 given in Çamtepe and Yener [2004]. Let us denote the set of keys assigned to nodes in the region $S_{i,j}$ by P_{ij}. Each of the regions has a distinct set of $p^2 + p + 1$ keys. So the entire size of the key pool is $r^2(p^2 + p + 1)$. Since $P_{i,j} \bigcap P_{i',j'} = 0$, for $(i, j) \neq (i', j')$, it can be ensured that even if a few nodes (or all nodes) within a region are compromised, none of the nodes (or link between nodes) in the other regions is affected. If two nodes share a common key then an *intralink* is said to exist between the nodes.

3.2 Modification of Çamtepe and Yener's Scheme

Modification of predistribution scheme. According to Çamtepe and Yener's [2004] scheme $p^2 + p + 1$ nodes are each preloaded with $p + 1$ keys according to a $PG(2, p)$, where p is a prime power. The construction of the symmetric design (which is the same as $PG(2, p)$) given by Çamtepe and Yener [2004] uses mutually orthogonal latin squares. They did not, however, provide an algorithm for shared key discovery and assumed that a shared key exists which can be found by the methods given in Eschenauer and Gligor [2002] and Chan et al. [2003]. Here we use a simpler construction (Algorithm 1) using the technique given in Street and Street [1987, Section 8.4]. This makes the shared key discovery algorithm much simpler. The complexity of our shared key discovery

algorithm is $O(1)$ and the communication overhead is $O(\log\ p)$ bits. We index the nodes (or blocks) by (a, b, c) where $a, b, c \in GF(p)$. The nodes are given by the identifiers $(1, b, c)$, $(0, 1, c)$, and $(0, 0, 1)$, where $b, c \in GF(p)$. So there are a total of $p^2 + p + 1$ nodes. Similarly the keys are indexed by (x, y, z) where $x, y, z, \in GF(p)$. The identifiers of the keys are given by $(x, y, 1)$, $(x, 1, 0)$, and $(1, 0, 0)$, where $x, y \in GF(p)$. So there are a total of $p^2 + p + 1$ keys (or elements). A key (x, y, z) is assigned to node (a, b, c) if $ax + by + cz = 0$. This design results in a $PG(2, p)$. For details one may refer to Street and Street [1987, Section 8.4]. We note that this predistribution is the same as that given by Çamtepe and Yener [2004]. However, this method is much simpler than calculating MOLS, then constructing affine planes, and then constructing projective planes as given in their construction [Çamtepe and Yener 2004, Section 3.1]. Steps 1–9 assign keys to nodes $(1, b, c)$, where $b, c \in GF(p)$. $(1, b, c)$ are assigned keys $(x, y, 1)$ such that $x + by + c = 0$. The key $(-b, 1, 0)$ is also assigned to $(1, b, c)$. Thus a total of $p + 1$ keys are assigned to $(1, b, c)$. Similarly $p + 1$ keys are assigned to the nodes $(0, 1, c)$ where $c \in GF(p)$ and $p + 1$ keys are assigned to the nodes $(0, 0, 1)$. Steps 1–9 will take $O(p^3) = O(n^{1.5})$ time, where n is the maximum number of nodes that the region can support. Steps 10–15 will take $O(p^2) = O(n)$. Steps 16–19 take $O(p) = O(\sqrt{n})$ time. Thus the algorithm takes $O(n^{1.5})$. This is the same as the algorithm given by Çamtepe and Yener [2007, Section III].

Shared key discovery algorithm. The shared key discovery algorithm is presented in Algorithm 2. All calculations are done modulo p.

Algorithm 1. Key Predistribution Using $PG(2, p)$

```
1: for Each element b in GF(p) do
2:   for Each element c in GF(p) do
3:     for Each element y in GF(p) do
4:       x = −(c + by)
5:       Assign key (x, y, 1) to node (1, b, c)
6:     end for
7:     Assign key (−b, 1, 0) to node (1, b, c)
8:   end for
9: end for
10: for Each element c in GF(p) do
11:   for Each element x in GF(p) do
12:     Assign key (x, −c, 1) to node (0, 1, c)
13:   end for
14:   Assign key (1, 0, 0) to node (0, 1, c)
15: end for
16: for Each element x in GF(p) do
17:     Assign key (x, 1, 0) to node (0, 0, 1)
18: end for
19: Assign key (1, 0, 0) to node (0, 0, 1)
```

Algorithm 2. Shared Key Discovery Using Symmetric Design

Require: : (a_i, b_i, c_i) and (a_j, b_j, c_j), the identifiers of nodes i and j respectively.
if $a_i = 0$ and $b_i = 0$ and $c_i = 1$ **then**
  if $a_j = 0$ and $b_j = 1$ **then**
    Identifier of the common key $= (1, 0, 0)$
  else
    Identifier of the common key $= (-b_j, 1, 0)$
  end if
else if $a_i = 0$ and $b_i = 1$ **then**
  if $a_j = 0$ and $b_j = 1$ **then**
    Identifier of the common key $= (1, 0, 0)$
  else
    Identifier of the common key $= (b_j c_i - c_j, -c_i, 1)$
  end if
else {When $(a_i, b_i, c_i) = (1, b_1, c_1)$ and $(a_j, b_j, c_j) = (1, b_2, c_2)$}
  Identifier of the common key $= (-c_1 + b_1 \frac{c_1 - c_2}{b_1 - b_2}, \frac{c_2 - c_1}{b_1 - b_2}, 1)$
end if

All the steps take $O(1)$ time to be calculated. The only information that needs to be broadcasted is the identifiers represented by the tuple (a, b, c), where a, b, c are in $GF(p)$. Thus the communication overhead is $O(\log\ p) = O(\log \sqrt{n})$, where n is the number of nodes in the region. This is very less compared to the time complexity given in Eschenauer and Gligor [2002] and Chan et al. [2003].

3.3 Key Predistribution in Agents Over the Entire Network

For any square region $S_{i,j}$ a set of three agents $a^1_{i,j}$, $a^2_{i,j}$, and $a^3_{i,j}$ are deployed. The set of $k + p + 1$ keys assigned to the three agents are denoted by $B^1_{i,j}$, $B^2_{i,j}$, and $B^3_{i,j}$. Apart from the $p + 1$ keys assigned from the set $P_{i,j}$, $B^1_{i,j}$ contains $\{(x, (xi + j) \bmod\ r) : 0 \leq x < k\}$, $B^2_{i,j}$ contains $\{(x, r + ((j - xi) \bmod\ r)) : 0 \leq x < k\}$, and $B^3_{i,j}$ contains $\{(x, 2r + ((xj + i) \bmod\ r)) : 0 \leq x < k\}$.

Consider the keys of the form (x, y), where $0 \leq x < k$. If $y < r$, then the keys are called Type I keys, if $r \leq y < 2r$, then the keys are called Type II keys, and if $y \geq 2r$, then the keys are called Type III keys. Refer to Table IV. We note that any agent contains keys of only one type. Depending on the type of keys an agent contains, agents may be of Type I, Type II, or Type III.

The agents can communicate with each other if they are within a fixed communication range. Generally this region around a given sensor node is the circular region with radius ρ called the RF radius and center as the sensor node itself. For simplicity we assume that agents within a particular region can communicate with agents which lie inside the Lee sphere (with Lee distance ρ) of that region (Lee sphere is defined earlier in this section). For this reason it is equivalent if we consider the three agents to be placed at the center of the region.

A natural question arises: Why not place one agent instead of three? The answer is that when nodes are compromised randomly, the probability that only one node is compromised is more than all the nodes are compromised. Had there been one agent in a region, two agents would have shared more than one key. In such a case a common key would have to be selected for communication. Now if this key is compromised, the shared key algorithm would have to be executed again to find a new common key. Since there are three agents and any two agents share at most one common key, if some key is compromised, the link is broken, there is no need of executing the shared key discovery algorithm again.

We could also have predistributed keys in such a way that nodes in two regions share some common keys. However, the compromise of one region adversely affects the other regions. Hence, though economical, this method is not proposed.

If two regions within Lee distance communicate via some common key belonging to some agent, then a *interlink* is said to exist. Each agent $B^l_{i,j}$ has an identifier (i, j, l) denoting the region $S_{i,j}$ where it belongs and l denoting the type of agent. To carry out secure communication, a common key needs to be established. Since we use a deterministic design having some definite structure, shared key discovery becomes very simple. Two agents of different type do not have any common key. However if two agents of the same type (say Type I) want to find out a shared key (if there exists one), then they broadcast their identifier. The common key will be obtained very easily using only an inverse calculation.

3.4 Shared Key Discovery

Suppose two Type I agents belonging to regions (i, j) and (i', j') want to find the shared key (if it exists). Then the shared key (x, y) will be such that $xi + j = xi' + j' \bmod r$. So if $x = (j' - j)(i - i')^{-1} < k$, then a common key exists. Suppose two Type II agents belonging to regions (i, j) and (i', j') want to find the shared key (if it exists). Then the shared key (x, y) will be such that $j - xi = j' - xi' \bmod r$. So if $x = (j' - j)(i' - i)^{-1} < k$, then a common key exists. Similarly to find if two Type II agents belonging to regions (i, j) and (i', j') want to find the shared key (if it exists), then the shared key (x, y) will be such that $xj + i = xj' + i' \bmod r$. So if $x = (i' - i)(j - j')^{-1} < k$, then a common key exists. The algorithm for shared key discovery runs in $O(1)$ time and only the identifier of the agent has to be sent. This results in a communication overhead of $O(\log r)$ bits. Since randomized key predistribution results in high communication and computation complexity, as discussed in Lee and Stinson [2008, Section 2.2.6], our approach is better than probabilistic key predistribution like Huang and Medhi [2007].

This can be found in constant time. If no common key exists, then a path key can be found using the technique given next.

Suppose node u belonging to region S_1 wants to communicate with node v belonging to region S_2, such that S_1 and S_2 are within the Lee distance each other. u generates a random key K and finds a common key, say k_1, it shares

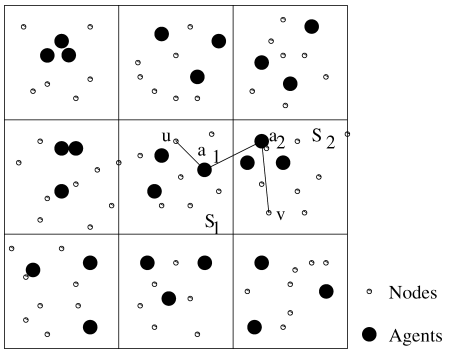


Fig. 2. A deployment region with nodes and agents. Communication between two nodes via agents is shown.

with one of the agents a_1 in S_1. It then sends the key K encrypted by k_1 to a_1. a_1 decrypts K using k_1. If a_1 shares a common key k_2 with agent a_2 of region S_2, then a_1 encrypts the key K with k_2 and sends it to a_2. a_2 then decrypts K using k_2 and sends it to v encrypted with the shared key k_3 between a_2 and v. v then decrypts K using k_3. So k is now a shared key (path key) between u and v. This is depicted in Figure 2. We next show that if $k \geq (r+1)/2$, then any two regions within Lee distance are connected.

THEOREM 3.1. *If $k \geq (r+1)/2$, then any two regions within Lee distance are connected via one or more common keys belonging to one or more agents.*

PROOF. Let us consider two regions $S_{i,j}$ and $S_{i',j'}$. If $B^1_{i,j} \bigcap B^1_{i',j'} \neq 0$, then $S_{i,j}$ is connected to $S_{i',j'}$. Suppose $B^1_{i,j} \bigcap B^1_{i',j'} = 0$, then for all $x = (j'-j)(i-i')^{-1} \bmod r \geq k$. Since $k \geq (r+1)/2$, $(j'-j)(i'-i)^{-1} < k-1$. For some x' if $j - x'i = j' - x'i' \bmod r$, then $x' < k-1$. So $B^2_{i,j}$ is connected to $B^2_{i',j'}$. If $i = i'$ then regions are not connected via Type I or Type II keys. However, all nodes where $i = i'$, are connected via Type III keys, because they share the key $(0, 2r+i)$. Hence all regions within the Lee distance of (i, j) are connected to it. □

Theorem 3.1 provides the justification of choosing only three agents for each region whatever be the size of the network.

Throughout the article we consider r to be prime and $k \geq (r+1)/2$.

4. ANALYSIS OF RESILIENCY

When sensor nodes are deployed in adversarial regions they are prone to be captured by enemies. When nodes are captured, the enemy gets access to the keys in it, so these compromised keys cannot be used for further communication. This may result in links between nodes being disrupted. In such cases communication needs to be carried by alternate paths. Sometimes nodes may be totally disconnected from the network. By node disconnectivity, we mean that the node is not physically disconnected from the network but logically disconnected from the network, meaning that the links with other nodes are broken. Similarly, when we say that a region is disconnected we mean that there is no

node which can communicate with the nodes in this region, because the links are disrupted. We consider selective node capture and random node capture models. We show that the selective node capture model cannot be mounted on our scheme.

4.1 Resiliency Against Selective Node Capture

During selective node capture the attacker compromises those nodes whose keys have not already been compromised. We note that any two nodes broadcast their node ids during the shared key discovery phase. The key identifiers are not broadcasted. Thus at no stage can the attacker know what key identifier is present in which node. Thus there is no way of knowing which nodes are left to be compromised. Thus unless the attacker compromises the node, she cannot choose a node for compromise to maximize the number of keys compromised. Hence our scheme is secure against selective node capture.

4.2 Random Node Compromise

We give measures of resiliency $E'(s)$, $E''(s)$, $V'(s)$, and $V''(s)$ as the proportion of links and nodes being broken, respectively. We consider two types of resiliency: local resiliency, which denotes the fraction of intralinks or nodes affected within a region and global resiliency, which denotes the fraction of interlinks and agents affected. Mathematically,

$$E'(s) = \frac{\text{Number of intralinks exposed after } s \text{ nodes are compromised}}{\text{Number of links present before compromised}}$$

$$E''(s) = \frac{\text{Number of interlinks exposed after } s \text{ agents are compromised}}{\text{Number of links present before compromised}}$$

$$V'(s) = \frac{\text{Number of nodes disconnected after } s \text{ nodes are compromised}}{\text{Number of nodes present before compromised}}$$

and

$$V''(s) = \frac{\text{Number of regions disconnected after } s \text{ agents are compromised}}{\text{Number of regions present before compromised}}.$$

We consider the local and global resiliency. By local resiliency we mean the resiliency within a particular region. This is measured in terms of $E'(s)$ (defined as the fraction of intralinks affected when s nodes are compromised) and $V'(s)$ (defined as the fraction of nodes disconnected when s nodes are compromised). By global resiliency we mean the resiliency of the entire region. This is measured in terms of $E''(s)$ (defined as the fraction of interlinks affected when s nodes are compromised) and $V''(s)$ (defined as the fraction of regions disconnected when s nodes are compromised). We give experimental and theoretical results for $E'(s)$, $E''(s)$, $V'(s)$, and $V''(s)$.

4.3 Estimation of $E'(s)$ and $E''(s)$

Let s sensor nodes be randomly compromised. Let s' nodes be compromised and s'' agents be compromised. We first find local resiliency $E'(s')$ (fraction

Table III. Experimental Value Vs Theoretical Value of $E'(s')$ When Number of Nodes in a Region is $n = p^2 + p + 1$, Keys Per Node is $p + 1$ and s' Nodes are Compromised

r	p	$n = p^2 + p + 1$	s'	$E'(s')$ (Experimental)	$E'(s')$ (Theoretical Upper Bound)
23	17	307	700	0.0336	0.0775
31	17	307	900	0.0534	0.0549
37	19	381	1200	0.0405	0.0460
37	23	553	2000	0.0616	0.0634
47	19	381	2000	0.0465	0.0475
47	23	553	2500	0.0680	0.0885

Experimental results are obtained for s' nodes chosen randomly over 100 runs.

of intralinks broken when s' nodes are compromised) and then calculate the global resiliency $E''(s'')$ (fraction of interlinks broken when s'' agents are compromised).

4.3.1 *Estimation of Local Resiliency $E'(s')$ for Intralinks.* According to the predistribution scheme any key within a particular region is present in exactly $p + 1$ sensor nodes, including the agents. Also, any two nodes have only one common key. So if a key K is compromised, then $p(p + 1)/2$ links are broken. Let us assume that k_i distinct keys are compromised from the S_i th region. Then $pk_i(p + 1)/2$ links are broken. Let us assume s_i nodes belonging to S_ith region are broken. We assume that all the keys exposed are distinct. This is an overestimate, because two compromised nodes may have a common key. So a maximum of $(p + 1)s_i$ keys are exposed. So a maximum of $s_i p(p + 1)^2/2$ intralinks are broken. Since there are only three agents, there are a maximum of 6 links broken between between the agents. We can ignore this. So the fraction of links broken when s_i nodes are compromised within region S_i is less than $\frac{s_i p(p+1)^2/2}{(p^2+p+1)(p^2+p)/2} = \frac{(p+1)s_i}{p^2+p+1}$. Suppose a total of s' nodes are compromised. These belong to the r^2 regions. Assuming that the compromised nodes are evenly distributed in the region, s'/r^2 nodes are nodes are broken per region. Hence the resiliency $E'(s') = \frac{1}{r^2} \sum_{i=0}^{r^2-1} \frac{(p+1)s_i}{p^2+p+1} = \frac{s'(p+1)}{r^2(p^2+p+1)}$.

The Table III gives the experimental and theoretical estimates of $E'(s')$.

4.3.2 *Estimation of Global Resiliency $E''(s'')$ for Interlinks.* We give an outline of how to compute the number of links broken when agents are compromised. The number of interlinks connected to each interior region is $2\rho(\rho + 1)$. The initial number of interlinks is less than $2r^2\rho(\rho + 1)$ (since the regions near the periphery will be connected to less number of regions). Let s'' agents be compromised. Any two regions in the same column are connected by Type III keys. Also any two regions not in the same row are connected by Type I and Type II keys. Any two regions share either one, two, or three keys of different types. Refer to Table IV. Since regions may be connected by more than one key, only if all the shared keys are compromised, the interlink is disrupted. We say that the triple $\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$ is a *good triple*, with (x_1, y_1) a Type I key, (x_2, y_2) a Type II key, (x_3, y_3) a Type III key, if whenever any one of the keys (x_i, y_i) occurs in an agent in a region, then (x_j, y_j), ($j \in \{1, 2, 3\}$ and $j \neq i$) also occurs in some other agents in the same region. For example, $\{(4, 4), (3, 11), (2, 15)\}$ is a good triple. Similarly, we say that the pair of keys

$\{(x_1, y_1), (x_2, y_2)\}$ is a *good pair* of Type I - Type II, with (x_1, y_1) a Type I key, (x_2, y_2) a Type II key, if whenever any one of the keys (x_i, y_i) occurs in a region, then (x_j, y_j), $(j \in \{1, 2\}$ and $j \neq i)$ also occurs in some agents in the same region but there is no key (x_3, y_3) of Type III such that $\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$ is not a good triple. Similarly we can define good pairs for Type II - Type III and Type I -Type III keys. For example, $\{(2, 3), (4, 19)\}$ is a good pair. However, $\{(4, 4), (3, 11)\}$ is not a good pair because $\{(4, 4), (3, 11), (2, 15)\}$ is a good triple. If there is a Type I or Type II or Type III key such that it does not form a good pair or good triple with some other type of key, then it is called an isolate. There are three types of isolates: Type I isolate, Type II isolate, and Type III isolate. For example, $(1, 7)$ is a Type II isolate in the example given in Table IV. There are no Type I isolates. We next find the conditions for existence of good triple or good pair. Let us consider two regions (i, j) and (i', j'). A good triple exists if all the three conditions that follows are satisfied.

$$x_1 i + j = x_1 i' + j' (\text{mod } r) \tag{2a}$$

$$-x_2 i + j = -x_2 i' + j' (\text{mod } r) \tag{2b}$$

$$x_3 j + i = x_3 j' + i' (\text{mod } r) \tag{2c}$$

From Eqs. (2a) and (2b) we find that $x_1 = -x_2 (\text{mod } r)$. Similarly, from Eqs. (2a) and (2c) we find that $x_1 = x_3^{-1} (\text{mod } r)$ and from Eqs. (2b) and (2c) we find that $x_2 = -x_3^{-1} (\text{mod } r)$. Hence a good triple exists for all $x_1, x_2, x_3 < k$, such that $x_1 = -x_2 = x_3^{-1}$. The good triple that arises when this condition holds is $\{(x_1, y_1), (-x_1, r + y_1), (x_1^{-1}, 2r + (x_1^{-1} y_1) \mod r))\}$. Similarly for good pairs consisting of Type I and Type II keys we have $x_1 = -x_2$, $x_1, x_2 < k$ and $x_1^{-1} \geq k$ and the good pair is given by $\{(x_1, y_1), (-x_1, r + y_1)\}$. For good pairs consisting of Type I and Type III keys we have $x_1 = x_3^{-1}$, $x_1, x_3 < k$ and $-x_1 \geq k$ and good pair of Type I-Type III is given by $\{(x_1, y_1), (x_1^{-1}, 2r + (x_1^{-1} y_1) \mod r))\}$. Similarly for good pairs consisting of Type II and Type III keys we have $x_2 = -x_3^{-1}$, $x_2, x_3 < k$ and $-x_2 \geq k$ and good pair of Type II-Type III is given by $\{(x_2, y_2), (-x_2^{-1}, 2r + (-x_2^{-1} y_2) \mod r))\}$.

Suppose s'' agents are compromised. We calculate the number of interlinks broken. We find all the distinct keys that are exposed. Generally when s'' agents are compromised, not all $3ks''$ keys are distinct, because two compromised agents may contain some common keys. Once we know the keys, we find the good triples and good pairs and isolates which are compromised. Then the number of interlinks broken within Lee distance ρ will be the number of interlinks connected by the compromised good triples, good pairs, and isolates. We explain this in Section 4.4.

4.4 Estimation of the Number of Links Disrupted When s'' Agents are Compromised

Suppose s'' agents are compromised. The compromised agents may be only of one type: Type I, Type II, or Type III, or may be a combination of these types. We enumerate the different conditions that can arise and how the links may be

Table IV. A 7×7 Region with 5 Keys in Each Agent

Region	Type I					Type II					Type III				
$S_{0,0}$:	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)	(0,7)	(1,7)	(2,7)	(3,7)	(4,7)	(0,14)	(1,14)	(2,14)	(3,14)	(4,14)
$S_{0,1}$:	(0,1)	(1,1)	(2,1)	(3,1)	(4,1)	(0,8)	(1,8)	(2,8)	(3,8)	(4,8)	(0,14)	(1,15)	(2,16)	(3,17)	(4,18)
$S_{0,2}$:	(0,2)	(1,2)	(2,2)	(3,2)	(4,2)	(0,9)	(1,9)	(2,9)	(3,9)	(4,9)	(0,14)	(1,16)	(2,18)	(3,20)	(4,15)
$S_{0,3}$:	(0,3)	(1,3)	(2,3)	(3,3)	(4,3)	(0,10)	(1,10)	(2,10)	(3,10)	(4,10)	(0,14)	(1,17)	(2,20)	(3,16)	(4,19)
$S_{0,4}$:	(0,4)	(1,4)	(2,4)	(3,4)	(4,4)	(0,11)	(1,11)	(2,11)	(3,11)	(4,11)	(0,14)	(1,18)	(2,15)	(3,19)	(4,16)
$S_{0,5}$:	(0,5)	(1,5)	(2,5)	(3,5)	(4,5)	(0,12)	(1,12)	(2,12)	(3,12)	(4,12)	(0,14)	(1,19)	(2,17)	(3,15)	(4,20)
$S_{0,6}$:	(0,6)	(1,6)	(2,6)	(3,6)	(4,6)	(0,13)	(1,13)	(2,13)	(3,13)	(4,13)	(0,14)	(1,20)	(2,19)	(3,18)	(4,17)
$S_{1,0}$:	(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(0,7)	(1,13)	(2,12)	(3,11)	(4,10)	(0,15)	(1,15)	(2,15)	(3,15)	(4,15)
$S_{1,1}$:	(0,1)	(1,2)	(2,3)	(3,4)	(4,5)	(0,8)	(1,7)	(2,13)	(3,12)	(4,11)	(0,15)	(1,16)	(2,17)	(3,18)	(4,19)
$S_{1,2}$:	(0,2)	(1,3)	(2,4)	(3,5)	(4,6)	(0,9)	(1,8)	(2,7)	(3,13)	(4,12)	(0,15)	(1,17)	(2,19)	(3,14)	(4,16)
$S_{1,3}$:	(0,3)	(1,4)	(2,5)	(3,6)	(4,0)	(0,10)	(1,9)	(2,8)	(3,7)	(4,13)	(0,15)	(1,18)	(2,14)	(3,17)	(4,20)
$S_{1,4}$:	(0,4)	(1,5)	(2,6)	(3,0)	(4,1)	(0,11)	(1,10)	(2,9)	(3,8)	(4,7)	(0,15)	(1,19)	(2,16)	(3,16)	(4,17)
$S_{1,5}$:	(0,5)	(1,6)	(2,0)	(3,1)	(4,2)	(0,12)	(1,11)	(2,10)	(3,9)	(4,8)	(0,15)	(1,20)	(2,18)	(3,18)	(4,14)
$S_{1,6}$:	(0,6)	(1,7)	(2,1)	(3,2)	(4,3)	(0,13)	(1,12)	(2,11)	(3,10)	(4,9)	(0,15)	(1,14)	(2,20)	(3,20)	(4,18)
$S_{2,0}$:	(0,0)	(1,2)	(2,4)	(3,6)	(4,1)	(0,7)	(1,12)	(2,10)	(3,8)	(4,13)	(0,16)	(1,16)	(2,16)	(3,16)	(4,16)
$S_{2,1}$:	(0,1)	(1,3)	(2,5)	(3,0)	(4,2)	(0,8)	(1,13)	(2,11)	(3,9)	(4,7)	(0,16)	(1,17)	(2,18)	(3,19)	(4,20)
$S_{2,2}$:	(0,2)	(1,4)	(2,6)	(3,1)	(4,3)	(0,9)	(1,7)	(2,12)	(3,10)	(4,8)	(0,16)	(1,18)	(2,20)	(3,15)	(4,17)
$S_{2,3}$:	(0,3)	(1,5)	(2,0)	(3,2)	(4,4)	(0,10)	(1,8)	(2,13)	(3,11)	(4,9)	(0,16)	(1,19)	(2,15)	(3,18)	(4,14)
$S_{2,4}$:	(0,4)	(1,6)	(2,1)	(3,3)	(4,5)	(0,11)	(1,9)	(2,7)	(3,12)	(4,10)	(0,16)	(1,20)	(2,17)	(3,14)	(4,18)
$S_{2,5}$:	(0,5)	(1,0)	(2,2)	(3,4)	(4,6)	(0,12)	(1,10)	(2,8)	(3,13)	(4,11)	(0,16)	(1,14)	(2,19)	(3,17)	(4,15)
$S_{2,6}$:	(0,6)	(1,1)	(2,3)	(3,5)	(4,0)	(0,13)	(1,11)	(2,9)	(3,7)	(4,12)	(0,16)	(1,15)	(2,14)	(3,19)	(4,19)
$S_{3,0}$:	(0,0)	(1,3)	(2,6)	(3,2)	(4,5)	(0,7)	(1,11)	(2,8)	(3,12)	(4,9)	(0,17)	(1,17)	(2,17)	(3,17)	(4,17)
$S_{3,1}$:	(0,1)	(1,4)	(2,0)	(3,3)	(4,6)	(0,8)	(1,12)	(2,9)	(3,13)	(4,10)	(0,17)	(1,18)	(2,19)	(3,20)	(4,14)
$S_{3,2}$:	(0,2)	(1,5)	(2,1)	(3,4)	(4,0)	(0,9)	(1,13)	(2,10)	(3,7)	(4,11)	(0,17)	(1,19)	(2,14)	(3,16)	(4,18)
$S_{3,3}$:	(0,3)	(1,6)	(2,2)	(3,5)	(4,1)	(0,10)	(1,7)	(2,11)	(3,8)	(4,12)	(0,17)	(1,20)	(2,16)	(3,19)	(4,15)
$S_{3,4}$:	(0,4)	(1,0)	(2,3)	(3,6)	(4,2)	(0,11)	(1,8)	(2,12)	(3,9)	(4,13)	(0,17)	(1,14)	(2,18)	(3,15)	(4,19)
$S_{3,5}$:	(0,5)	(1,1)	(2,4)	(3,0)	(4,3)	(0,12)	(1,9)	(2,13)	(3,10)	(4,7)	(0,17)	(1,15)	(2,20)	(3,18)	(4,16)
$S_{3,6}$:	(0,6)	(1,2)	(2,5)	(3,1)	(4,4)	(0,13)	(1,10)	(2,7)	(3,11)	(4,8)	(0,17)	(1,26)	(2,15)	(3,14)	(4,20)
$S_{4,0}$:	(0,0)	(1,4)	(2,1)	(3,5)	(4,2)	(0,7)	(1,10)	(2,13)	(3,9)	(4,12)	(0,18)	(1,18)	(2,18)	(3,18)	(4,18)
$S_{4,1}$:	(0,1)	(1,5)	(2,2)	(3,6)	(4,3)	(0,8)	(1,11)	(2,7)	(3,10)	(4,13)	(0,18)	(1,19)	(2,20)	(3,14)	(4,15)
$S_{4,2}$:	(0,2)	(1,6)	(2,3)	(3,0)	(4,4)	(0,9)	(1,12)	(2,8)	(3,11)	(4,7)	(0,18)	(1,20)	(2,15)	(3,17)	(4,19)
$S_{4,3}$:	(0,3)	(1,0)	(2,4)	(3,1)	(4,5)	(0,10)	(1,13)	(2,9)	(3,12)	(4,8)	(0,18)	(1,14)	(2,17)	(3,20)	(4,16)
$S_{4,4}$:	(0,4)	(1,1)	(2,5)	(3,2)	(4,6)	(0,11)	(1,7)	(2,10)	(3,13)	(4,9)	(0,18)	(1,15)	(2,19)	(3,16)	(4,20)
$S_{4,5}$:	(0,5)	(1,2)	(2,6)	(3,3)	(4,0)	(0,12)	(1,8)	(2,11)	(3,7)	(4,10)	(0,18)	(1,16)	(2,14)	(3,19)	(4,17)
$S_{4,6}$:	(0,6)	(1,3)	(2,0)	(3,4)	(4,1)	(0,13)	(1,9)	(2,12)	(3,8)	(4,11)	(0,18)	(1,17)	(2,16)	(3,15)	(4,14)
$S_{5,0}$:	(0,0)	(1,5)	(2,3)	(3,1)	(4,6)	(0,7)	(1,9)	(2,11)	(3,13)	(4,8)	(0,19)	(1,19)	(2,19)	(3,19)	(4,19)
$S_{5,1}$:	(0,1)	(1,6)	(2,4)	(3,2)	(4,0)	(0,8)	(1,10)	(2,12)	(3,7)	(4,9)	(0,19)	(1,20)	(2,14)	(3,15)	(4,16)
$S_{5,2}$:	(0,2)	(1,0)	(2,5)	(3,3)	(4,1)	(0,9)	(1,11)	(2,13)	(3,8)	(4,10)	(0,19)	(1,14)	(2,16)	(3,18)	(4,20)
$S_{5,3}$:	(0,3)	(1,1)	(2,6)	(3,4)	(4,2)	(0,10)	(1,12)	(2,7)	(3,9)	(4,11)	(0,19)	(1,15)	(2,18)	(3,14)	(4,17)
$S_{5,4}$:	(0,4)	(1,2)	(2,0)	(3,5)	(4,3)	(0,11)	(1,13)	(2,8)	(3,10)	(4,12)	(0,19)	(1,16)	(2,20)	(3,17)	(4,14)
$S_{5,5}$:	(0,5)	(1,3)	(2,1)	(3,6)	(4,4)	(0,12)	(1,7)	(2,9)	(3,11)	(4,13)	(0,19)	(1,17)	(2,15)	(3,20)	(4,18)
$S_{5,6}$:	(0,6)	(1,4)	(2,2)	(3,0)	(4,5)	(0,13)	(1,8)	(2,10)	(3,12)	(4,7)	(0,19)	(1,18)	(2,17)	(3,16)	(4,15)
$S_{6,0}$:	(0,0)	(1,6)	(2,5)	(3,4)	(4,3)	(0,7)	(1,8)	(2,9)	(3,10)	(4,11)	(0,20)	(1,20)	(2,20)	(3,20)	(4,20)
$S_{6,1}$:	(0,1)	(1,0)	(2,6)	(3,5)	(4,4)	(0,8)	(1,9)	(2,10)	(3,11)	(4,12)	(0,20)	(1,14)	(2,15)	(3,16)	(4,17)
$S_{6,2}$:	(0,2)	(1,1)	(2,0)	(3,6)	(4,5)	(0,9)	(1,10)	(2,11)	(3,12)	(4,13)	(0,20)	(1,15)	(2,17)	(3,19)	(4,14)
$S_{6,3}$:	(0,3)	(1,2)	(2,1)	(3,0)	(4,6)	(0,10)	(1,11)	(2,12)	(3,13)	(4,7)	(0,20)	(1,16)	(2,19)	(3,15)	(4,18)
$S_{6,4}$:	(0,4)	(1,3)	(2,2)	(3,1)	(4,0)	(0,11)	(1,12)	(2,13)	(3,7)	(4,8)	(0,20)	(1,17)	(2,14)	(3,18)	(4,15)
$S_{6,5}$:	(0,5)	(1,4)	(2,3)	(3,2)	(4,1)	(0,12)	(1,13)	(2,7)	(3,8)	(4,9)	(0,20)	(1,18)	(2,16)	(3,14)	(4,19)
$S_{6,6}$:	(0,6)	(1,5)	(2,4)	(3,3)	(4,2)	(0,13)	(1,7)	(2,8)	(3,9)	(4,10)	(0,20)	(1,19)	(2,18)	(3,17)	(4,16)

affected due to these conditions. As discussed in Section 4, not all exposed keys are distinct. We also note that only agents of the same type can have keys in common. For example, agent of Type I can share at most one key with another agent of Type I, but none of Type II or Type III. Given an existing interlink

we can say if it will be disconnected if all the shared keys in all the agents are exposed. There may be one, two, or three shared keys between an interlink. Only if all the shared keys are exposed, the link is broken. For example regions $S_{1,3}$ and $S_{3,2}$ have three keys (4, 0), (3, 7), and (2, 14) in common. So if all three keys are compromised, then the link $S_{1,3}S_{3,2}$ will be broken, provided they are within communication range. Again we note that {(4, 0), (3, 7), (2, 14)} is a good triple. So whenever agents are compromised, we check if good triples and good pairs are compromised. Then the interlinks connected by these triples and pairs will be broken. Also if there are isolates which are exposed, then interlinks which are connected by these isolates are broken. We illustrate these different cases with the example given in Table IV. $L(x, y)$ denotes the number of interlinks within communication range that are connected by key (x, y).

(1) If all the compromised agents are of Type I, then number of links broken is given by

$$\sum_{(x,y)} L(x, y) \text{ , where } (x, y) \text{ is a compromised isolate of Type I.}$$

Since Type II and Type III keys are not compromised, no good triple or good pair is compromised. Only the interlinks which are connected via compromised Type I isolates will be broken. In the example given, since there are no isolates of Type I, no links will be compromised if only Type I agents are compromised. All interlinks will be connected by Type II or Type III keys.

(2) If all the compromised agents are of Type II, then number of links broken is given by

$$\sum_{(x,y)} L(x, y - r) \text{ , where } (x, y) \text{ is a compromised isolate of Type II.}$$

The reason is same as Case 1. For example, if Type II agents in $S_{2,1}$ and $S_{4,5}$ are compromised, then the keys (0, 8), (1, 13), (2, 11), (3, 9), (4, 7), (0, 12), (1, 8), (3, 7), and (4, 10) are exposed. Only (1, 13) and (1, 8) are isolates of Type II and all the interlinks within communication range which are connected by these two keys will be broken. No interlinks connected by (0, 8) will be broken because {(0, 8), (0, 1)} is a good pair and (0, 1) is not compromised. So all interlinks which were connected via (0, 8) will still be connected by (0, 1). Similar is the case with other exposed keys.

(3) If all the compromised agents are of Type III, then number of interlinks broken is given by

$$\sum_{(x,y)} L(x, y - 2r) \text{ , where } (x, y) \text{ is a compromised isolate Type of III.}$$

The reason is same as Case 1. For example, if Type III agents in $S_{1,6}$ and $S_{4,5}$ are compromised, then the keys (0, 15), (1, 14), (2, 20), (3, 19), (4, 18), (0, 18), (1, 16), (2, 14), and (4, 17) are exposed. Only (0, 15) and (0, 18) are isolates of Type III and all the interlinks within communication range which are connected by these two keys will be broken. No other interlinks will be broken since good pairs and good triples exist which are not fully exposed.

(4) If all the compromised agents are of Type I or Type II, then number of interlinks broken is given by

$\sum_{(x,y)} L(x, y)$, where $\{(x, y), (-x, r + y)\}$ is a compromised good pair +
$\sum_{(x,y)} L(x, y)$, where (x, y) is a Type I compromised isolate +
$\sum_{(x,y)} L(x, y - r)$, where (x, y) is a Type II compromised isolate.

For example, if Type I agent of region $S_{2,0}$ and Type II agent of region $S_{3,3}$ are compromised, then the keys $(0, 0)$, $(1, 2)$, $(2, 4)$, $(3, 6)$, $(4, 1)$, $(0, 10)$, $(1, 7)$, $(2, 11)$, $(3, 8)$, and $(4, 12)$ are exposed. $\{(4, 1), (3, 8)\}$ is not a compromised good pair, since $\{(4, 1), (3, 8), (2, 16)\}$ is a good triple and $(2, 16)$ is not exposed. So the interlink connected by $(4, 1)$ will not be compromised. $(1, 7)$ is a isolate and so the interlinks connected by $(1, 7)$ will be broken. None of the other links will be broken because all other keys belong to some good pair or triple which have not been fully exposed. If the Type I agent in region $S_{2,0}$ and Type II agent of region $S_{3,4}$, then the interlinks connected by $(3, 6)$ (and $(4, 13)$)and $(1, 8)$ will be broken. The number of interlinks broken is $L(3, 6) + L(1, 1)$.

(5) If all the compromised agents are of Type I or Type III, then number of interlinks broken is given by

$\sum_{(x,y)} L(x, y)$, where $\{(x^{-1}, 2r + x^{-1}y \mod r)$ is a compromised good pair
$+ \sum_{(x,y)} L(x, y)$, where (x, y) is a Type I compromised isolate
$+ \sum_{(x,y)} L(x, y - 2r)$, where (x, y) is a Type III compromised isolate.

(6) If all the compromised agents are of Type II or Type III, then the number of interlinks broken is given by

$\sum_{(x,y)} L(x, y - r)$, where $(x, y), (r + (-x^{-1}) \bmod r, 2r + (-x^{-1}y) \mod r)\}$
is a compromised good pair +
$\sum_{(x,y)} L(x, y - r)$, where (x, y) is a Type II compromised isolate +
$\sum_{(x,y)} L(x, y - 2r)$, where (x, y) is a Type III compromised isolate.

(7) If the compromised agents are of Type I, Type II, or Type III, then the number of interlinks broken is given by

$\sum_{(x,y)} L(x, y)$, where $\{(x, y), (-x, r + y), (x^{-1}, 2r + (x^{-1}y) \mod r)\}$ is a
compromised good triple +
$\sum_{(x,y)} L(x, y)$, where $\{(x, y), (-x, r + y)\}$ is a compromised good pair +
$\sum_{(x,y)} L(x, y)$, where $\{(x, y), (x^{-1}, 2r + (x^{-1}y) \mod r)\}$ is a compromised
good pair +
$\sum_{(x,y)} L(x, y - r)$, where $\{(x, y), (-x^{-1}, 2r + (-x^{-1}y) \mod r)\}$ is a
compromised good pair +
$\sum_{(x,y)} L(x, y)$, where (x, y) is a compromised isolate of Type I +
$\sum_{(x,y)} L(x, y - r)$, where (x, y) is a compromised isolate of Type II +
$\sum_{(x,y)} L(x, y - 2r)$, where (x, y) is a compromised isolate of Type III.

For example, let Type I agent of $S_{2,3}$, Type II agents of $S_{0,4}$ and $S_{5,3}$ and Type III agent of $S_{4,2}$, are compromised. Then the keys $(0, 3)$, $(1, 5)$, $(2, 0)$, $(3, 2)$, $(4, 4)$, $(0, 11)$, $(1, 11)$, $(2, 11)$, $(3, 11)$, $(4, 11)$, $(0, 10)$, $(1, 12)$, $(2, 7)$, $(3, 9)$, $(0, 18)$, $(1, 20)$, $(2, 15)$, $(3, 17)$, $(4, 19)$ are exposed. Since $\{(4, 4), (3, 11), (2, 15)\}$ is a good triple and is fully compromised, the interlinks connected by $(4, 4)$ will be broken. $(0, 3), (0, 10)$ is a good pair and fully compromised, so the

interlink connected by $(0, 3)$ will be broken. Links connected by Type II isolates $(1, 11)$ and $(1, 12)$ and Type III isolates of $(0, 18)$ will also be broken. None of the other links will be broken because they belong to some pair or triple which are not fully compromised. The number of interlinks broken will be given by $L(4, 4) + L(0, 3) + L(1, 4) + L(1, 5) + L(0, 4)$.

Since we know how the interlinks will be affected we calculate the number of interlinks connected by key (x, y). This number is denoted by $L(x, y)$. A region S is called an *interior region* if the Lee sphere of radius ρ surrounding S contains an agent of the network. For simplicity we calculate all the links that are connected to interior region S via (x, y). Let (x, y) be a Type I key. Suppose a Type I agent belong to the region $S_{i,j}$ which has been compromised. Let it contain key (x, y). We find the regions $S_{i',j'}$ within Lee distance ρ of $S_{i,j}$ which share the key (x, y). If two regions $S_{i,j}$ and $S_{i-t,j'}$ share the key (x, y), then $xi + j = x(i - t) + j' (\mathrm{mod}\ r)$. So $j' = j + tx (\mathrm{mod}\ r)$. Thus we look at the regions $S_{i-1,j+x}, S_{i-2,j+2x}, \cdots, S_{i-t,j+tx}$ and $S_{i+1,j-x}, S_{i+2,j-2x}, \cdots, S_{i+t,j-tx}$ share key (x, y) with the region $S_{i,j}$. We want to consider only those regions which lie within the Lee sphere of radius ρ. So $|t| \leq \rho$ and either $tx \bmod r \leq \rho - t$ or $r - |tx \bmod r| \leq \rho - t$. Since x is known, the number of regions sharing key (x, y) within the communication range is the same as finding the number of values of t which satisfy the equations

$$|t| \leq \rho \; and \; |tx \bmod r| \leq \rho - t \tag{3a}$$

and

$$|t| \leq \rho \; and \; r - |tx \bmod r| \leq \rho - t. \tag{3b}$$

For simplicity, we consider only interior regions and the number of regions connected to a particular region is comprised of the solutions of t of the following equation.

$$t \leq \rho \; and \; tx \bmod r \leq \rho - t \tag{4}$$

We now consider agents of Type II and Type III. If two regions $S_{i,j}$ and $S_{i-t,j'}$ share the Type II key $(x, r + y)$, then $-xi + j = -x(i - t) + j' (\mathrm{mod}\ r)$. So $j' = j - tx (\mathrm{mod}\ r)$. Thus we see that the regions $S_{i-1,j-x}, S_{i-2,j-2x}, \cdots, S_{i-t,j-tx}$ and $S_{i+1,j+x}, S_{i+2,j+2x}, \cdots, S_{i+t,j+tx}$ are the share key $(x, r + y)$ with the region $S_{i,j}$. We want to consider only those regions which lie within the Lee sphere of radius ρ. Since we consider only the interior regions, the number of interlinks connected by key $(x, r + y)$ of Type II we need to find solutions to Eq. (4). For Type III agents if two regions $S_{i,j}$ and $S_{i',j-t}$ share the Type III key $(x, 2r + y)$, then $xj + i = x(j - t) + i' (\mathrm{mod}\ r)$. So $i' = i + tx (\mathrm{mod}\ r)$. Thus we see that the regions $S_{i+x,j-1}, S_{i+2x,j-2}, \cdots, S_{i+tx,j-t}$ and $S_{i-x,j+1}, S_{i-2x,j+2}, \cdots, S_{i-tx,j+t}$ are the share key $(x, 2r + y)$ with the region $S_{i,j}$. We want to consider only those regions which lie within the Lee sphere of radius ρ. Since we consider only the interior regions the number of interlinks connected by key $(x, 2r + y)$ of Type III we need to find solutions to Eq. (4).

To find the number of regions connected by Type II key $(x, r + y)$ we find $L(x, y)$ and the number of regions connected by Type III key $(x, 2r + y)$ we find $L(x, y)$. The same formula holds for Type II and Type III keys.

Table V. Experimental Value of $E''(s'')$ When Number of Regions is r^2, Number of Keys in Each Agent is k and the Lee Distance is ρ and s'' Agents are Compromised

r	k	ρ	s''	$E''(s'')$
23	15	7	10	0.04589
23	15	5	10	0.05317
31	20	7	10	0.04082
37	30	7	40	0.11696
47	40	9	50	0.09948
53	50	10	50	0.07038

Experimental results are obtained for s'' agents chosen randomly over 100 runs.

When s'' agents are compromised randomly, number of interlinks disrupted cannot be calculated deterministically. This is because of the following reasons.

(1) The exact number of distinct keys cannot be deterministically calculated. This is because it depends on the position of the agents over the entire region.
(2) The fraction of links compromised depends on the number of good triples, good pairs, and isolates which are compromised. Existence of these triples, pairs, and isolates depends on the existence of inverse which are less than k. Since inverses are randomly scattered over r, it is difficult to find the number of triples, pairs, or isolates affected.

The experimental results for $E''(s'')$ for interlinks is given in Table V.

4.5 Estimation of $V'(s)$ and $V''(s)$

When sensor nodes are compromised, keys are exposed. There is chance that all the keys in a noncompromised node are exposed. So this node is totally disconnected from the network. When links are broken, communication can occur through alternate paths. However, when sensor nodes are disconnected no communication with the disconnected nodes is possible. Hence node loss must be prevented. None of the other papers on key predistribution using deployment knowledge has discussed this very important issue so far. We look at the local resiliency $V'(s)$ (defined as the fraction of nodes disconnected when s nodes are compromised) and global resiliency $V''(s)$ (defined as the fraction of regions disconnected when s agents are compromised). Our scheme reduces the chance of sensor nodes being disconnected altogether. To disconnect a node all the keys contained therein are exposed. Since a node n contains $p+1$ keys, a minimum of $p+1$ nodes having those keys must be compromised to disconnect a node in that region. Suppose s' nodes are compromised. An average of s'/r^2 nodes are compromised in each region. So $s'/r^2 > p+1$. This means on an average $s' > r^2(p+1)$ nodes have to be compromised to disconnect one node. For example, for a 31×31 grid, if nodes in each region contain 18 keys ($p = 17$), then on an average 17298 nodes must be compromised to disconnect one node of each

Table VI. Experimental Value of $V''(s'')$ When Number of Regions is r^2, Number of Keys per Agent is k and s'' Agents are Compromised

r	k	s''	$V''(s'')$
23	15	100	0.00028
31	20	200	0.0027
37	30	300	0.0067
47	40	400	0.0022
53	50	500	0.0043

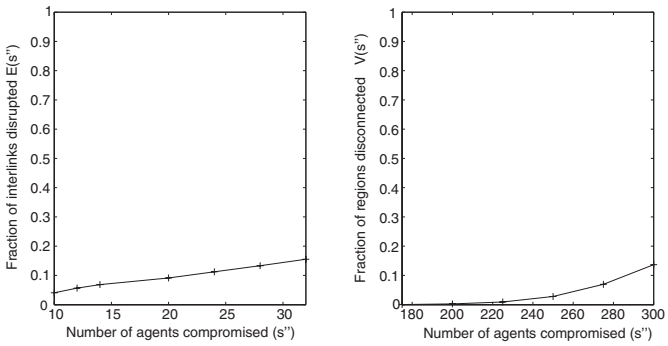
Experimental results are obtained for s'' agents chosen randomly over 100 runs.

region. This value of s' is quite large. So we can say that our scheme is very resilient. Only very few nodes will be disconnected even when a large number of nodes are compromised.

We next find the number of regions disconnected when s'' agents are compromised. We give experimental results in Table VI. From the table we note that for a 31×31 grid if 200 agents are compromised then about $2 - 3$ regions will be disconnected, where each agent contains 20 keys. When agents contain 25 keys then on compromising 200 agents about one region will be disconnected. The theoretical bound for $V''(s'')$ is difficult to estimate because it depends on the position and type of agents compromised. The position and type of agents determine which keys are being exposed. If all distinct keys are exposed, then $V''(s'')$ is expected to be more than the case where there is an overlap between the exposed keys. Also if the keys exposed form a good triple or good pair, then $V''(s'')$ is expected to be more than if one of the keys belonging to a good triple or good pair is exposed. Further, since finding good triple or good pair depends on the existence of inverse within a certain range ($< k$), the problem becomes all the more difficult, as we know that inverse are scattered randomly. The experimental results for $E''(s'')$ and $V''(s'')$ are given in Figure 3.

5. RELATED WORK

Key predistribution using deployment knowledge has been studied in Du et al. [2004, 2006], Yu and Guan [2005, 2008], Liu and Ning [2003, 2005], Younis et al. [2006], Zhou et al. [2006], Huang et al. [2004], Huang and Medhi [2007], Chan and Perrig [2005], and Simonova et al. [2006]. Blom [1984] proposed a method to predistribute keys using symmetric matrices. In this scheme there are two matrices, a public matrix G which is known to all the nodes and a private matrix D. A symmetric matrix $A = (a_{ij})$ is constructed as $(D.G)^T$, where $(D.G)^T$ is the transpose of the matrix $D.G$. The ijth entry is $a_{ij} = a_{ji}$ is a pairwise key between the ith and jth nodes. Each node i stores a corresponding row of the private matrix and is able to compute a pairwise key with any other node j by multiplying the row of the private matrix it stores with the corresponding column of the public matrix. Du et al [2003, 2005] proposed a multiple-space Blom scheme and in which there are multiple key spaces. In this scheme first w key spaces are generated and each sensor is loaded with τ key spaces, $2 \leq \tau \leq w$.



(a) fraction of interlinks affected $E''(s'')$ when s'' agents are compromised. $r = 31, k = 20$

(b) fraction of regions disconnected $V''(s'')$ when s'' agents are compromised. $r = 31$, $k = 20$

Fig. 3. Study of resiliency.

This further improved resilience of the scheme. An improvement of this scheme was made using deployment knowledge in Du et al. [2004, 2006].

Blom [2004] scheme was also used by Huang et al. [2004] and Huang and Medhi [2007] in devising a location-aware scheme for WSN. This scheme is secure against selective node capture attack.

Yu and Guan [2005, 2008] studied key predistribution schemes using deployment knowledge and compared the effect of deployment on triangular, hexagonal, and square grids. They show that hexagonal, grids give better connectivity and resiliency than triangular and square grids. They make use of Blom's [1984] scheme for key predistribution. Their schemes have low storage and full connectivity between nodes within the communication range. If the number of nodes compromised is less than some threshold value, then the communication between any other nodes is secure.

According to their scheme, the entire deployment area S_f is broken down into t grids equally, where shapes of grids may be various. N nodes are also divided equally into t groups; group i is deployed in grid i. The center of the grid is a deployment point, which is the desired location of a group of nodes. It is assumed that the location of the nodes of each group i follows some probability distribution function (pdf) $f_i(x, y) = f(x, y, \mu_{x_i}, \mu_{y_i})$, where $(\mu_{x_i}, \mu_{y_i}) \in S_f$ is the coordinate of the deployment point of the group.

In Blundo et al. [1998] the authors proposed polynomial-based key generation. It uses polynomial evaluations to obtain a pairwise key. Each node i gets polynomial shares $f(i, x)$ of symmetric polynomials of degree λ. In order to calculate a common key with node j, node i needs to evaluate its polynomial with at $x = j$ as $f(i, j)$. Node j would in turn evaluate $f(j, i)$ and since polynomials are symmetric $f(i, j) = f(j, i)$ this value could be used as a common key. This scheme is resistant against node capture: if less than $\lambda + 1$ nodes are compromised, no information about the keys is revealed. This method has been used by Liu and Ning [2003, 2005] to distribute keys in sensor nodes. Apart from

using Blundo's [1998] scheme Liu and Ning's [2003, 2005] scheme used deployment knowledge. According to their scheme, the entire deployment region is broken into rectangular regions. The deployment region is broken down into equal-sized squares $\{C_{i_c,i_r}\}_{i_c=0,1,\ldots,C-1,i_r=0,1,\ldots,R-1}$, 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 $s = R \times C$ grid the setup server generates s 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}.

In Zhou et al. [2006] the authors discussed a key predistribution scheme where sensor nodes are mobile. There are static sensors which are deployed in groups. There are mobile collectors which are used used to collect and aggregate sensor data and forward to the base station. For key predistribution it is assumed that there are n_s sensor nodes and n_m mobile collectors. The static sensors are arranged in g groups G_i, $1 \leq i \leq g$, each of which has $\gamma = n_s/g$ sensors. Group G_u will comprise sensors s_i such that $(u-1)\gamma < i \leq u\gamma$. The pairwise key between a pair of sensor nodes is denoted by $S - S$ key and the pairwise key between a mobile collector and a sensor as an $M - S$ key. All senor nodes within a group are connected to each other using pairwise keys. If sensors s_i and s_j belong to the same group they start off associated. If they are in different groups then communication takes place by agents. Groups G_u and G_v are are said to be t-associated if they share t agents for each other. Each pair of agents shares a pairwise key.

In Simonova et al. [2006], the authors discuss two predistribution schemes, one using deployment knowledge: one in which the network is homogeneous and the second in which the network is heterogeneous. In both schemes the entire deployment region is broken down into grids. Sensors are deployed in groups in these grids similar to Du et al. [2006]. There are two types of key pools: the original key pool and the deployment key pool. Each grid has a different original key pool. All the original key pools corresponding to the different grids are disjoint. Let the original key pool belonging to the cell $h_{i,j}$ be denoted by $OKP_{i,j}$. Initially keys in the sensors are distributed from the original key pool and placed in the sensors. Then the deployment regions are grouped together into bigger cells, each having m^2 cells. For a deployment cell $h_{i,j}$ the original key pools from the m^2 cells are merged to form the deployment key pool. For the cell $h_{i,j}$ the corresponding deployment key pool is denoted by $DKP_{i,j} = \{\bigcup_{x,y} OKP_{x,y} | x = 1, 2, \ldots, (i+m), y = 1, 2, \ldots, (j+m)\}$. So the grid is augmented by $m-1$ cells on the horizontal and vertical sides before the key pools are constructed. Simonova et al. [2006] state that any key predistribution can be used to assign keys from the key pool. However, they consider the transversal designs used by Lee and Stinson [2008].

The second scheme is heterogeneous in which there are two types of nodes: the strong nodes and the weak nodes. The strong nodes contain a greater number of keys (mem_s keys in each sensor), compared to the weak nodes (mem_w keys in each sensor) and have higher communication range than the weak nodes. Initially mem_w keys are distributed in each of the weak nodes. Once the weak nodes are deployed, the cells are grouped together as supercells. The $mem_s - mem_w$

Table VII. Comparison of the Different Schemes with Respect to the Type of Deployment, Type of Nodes - Nodes of Same Strength(homogeneous) or Different Strength(heterogeneous), Communication Overhead, Storage and Scalability

Schemes	Deployment	Nodes	Communication cost	Storage	Scalability
DDHV [Du et al. 2004; Du et al. 2006]	Grid-group	Homogeneous	$O(\tau)$	$\tau(\lambda+1)$	Scalable
LN [Liu and Ning 2003, 2005]	Grid	Homogeneous	$O(\log C \log R)$	$(t+1)q$	Not scalable
YG [Yu and Guan 2005, 2008]	Grid-group	Homogeneous	$\lambda(\log g)$	$(\lambda+1)\omega$	Not scalable
ZNR [Zhou et al. 2006]	Group	Heterogeneous	$O(\log N)$	$O(\gamma)^1$ $O(n_s)^2$	Not scalable
HMMH [Huang et al. 2004]	Grid-group	Homogeneous	$O(\tau)$	$\tau(\lambda+1)$	Scalable
HM [Huang and Medhi 2007]	Grid-group	Homogeneous	$O(\tau)$	$\tau(\lambda+1)$	Scalable
PIKE [Chan and Perrig 2005]	Grid	Homogeneous	$O(\log\sqrt{N})$	$O(\sqrt{N})$	Not Scalable
SLW [Simonova et al. 2006]-1	Grid-group	Homogeneous	$O(\log p')$	$O(\sqrt{N/g})$	Scalable
SLW [Simonova et al. 2006]-2	Grid-group	Heterogeneous	$O(\log p')$	$O(\sqrt{N/g})$	Scalable
Ours	Grid-group	Heterogeneous	$O(\log p)$	$O(\log n)^1$ $O(\log N)^2$	Not Scalable

Here τ denotes the number of key spaces selected out of ω spaces in DDHV scheme, λ denotes the security parameter for the Blom scheme, ω denotes the number of key spaces, t denotes the degree of polynomial whose coefficients are in F_q. $C \times R$ is the area of the region for LN scheme. g is the number of groups, n is the number of nodes in each group and $\gamma = n_s/g$, where n_s is the total number of sensors. N is the total number of sensors. p and p' are parameters in our scheme and that of SLW schemes respectively. [1] is the storage for small sensor nodes and [2] is the storage for agents.

Table VIII. Comparative Study of Intraconnectivity with the Number of Keys

Schemes	Number of keys	Connectivity
DDHV [Du et al. 2004; Du et al. 2006]	200	0.92
LN [Liu and Ning 2003, 2005]	200	0.99
ZNR [Zhou et al. 2006]	100	1.00
SLW [Simonova et al. 2006]-1	16	0.5856
SLW [Simonova et al. 2006]-2	40	0.80
Ours	12	1.00

The size of the network in DDHV,LN,ZNR is 10000, for SLW it is 12100 and 16093 for our scheme.

keys are now placed in the strong nodes. The rest of the predistribution scheme is similar to the one given in the previous scheme.

5.1 Comparison with Other Schemes

We present a comparative study of communication, storage, and scalability of several schemes in Table VII.

The scheme given by Huang et al. [2004] (HMMH) and Huang and Medhi [2007] (HM) differ from our scheme in that the sensor nodes are randomly

distributed in a two-dimensional field which is divided into rectangular regions. All nodes have equal power and storage capacities (homogeneous) compared to our scheme where there are two different types of nodes. The nodes in a region can communicate directly with each other with probability > 0.5. Our scheme is better in the respect that all nodes in a region can directly communicate with each other, thus reducing delays in communication. Since shared key discovery is done by matching identifiers of keys in the sensor nodes, which will involve huge computation and communication costs. We consider a network with a total of $10,000$ nodes. There are 100 regions, each having 100 nodes. Suppose 300 nodes are compromised, then fraction of links compromised amongst uncompromised nodes is negligible in Huang et al. [2004] and Huang and Medhi [2007]. However, if all the nodes are considered then the fraction of links compromised will be higher. With $16,093$ sensor nodes, distributed in 121 regions of 133 nodes each, the fraction of intralinks broken is about 0.07. However, this includes not just the links between the uncompromised sensor nodes but all the compromised and uncompromised sensor nodes.

We next compare our scheme with that given by Zhou et al. [2006] (ZNH). The sensor nodes are deployed similarly as in our scheme; however, their scheme employs mobile agents instead of static agents as in our case. The nodes in a region have 99 keys, which is only 12 in our case, for a region containing 100 nodes. In their scheme resiliency is defined to be the fraction of links broken, where links are said to exist between two nodes provided they have a common key or a path key which is established between nodes where there is no direct communication. As such, the resiliency will be lower than ours.

Deployment knowledge is also employed by Liu and Ning [2003, 2005] (LN) and Blackburn et al. [2008] (BEMP). There the whole region is divided into squares as in our scheme, but instead of a group of nodes being deployed in a square as in our scheme, only one node is placed in a square in these schemes.

Though deployment knowledge has been used by Younis et al. [2006] (YGE) and Du et al. [2006] (DDHV), the deployment scheme is different in that there are no specialized agents to communicate between regions. In these schemes direct communication between nodes in not guaranteed.

In the Simonova et al. [2006] scheme, the number of specialized nodes depends upon the size of the network, unlike ours which is constant ($= 3$). 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.

Though the number of groups is chosen to be r^2, where r is a prime power, our design works in all those cases where the dimension n of the grid is not a prime. This can be done by simply choosing a prime power $r > n$ and neglecting those regions which fall out of the $n \times n$ grid.

Our scheme has several advantages. Firstly, the number of keys per node is very low, of the order of $\sqrt{n}$, where n is the number of nodes in a region. Secondly, the agents also have $\sqrt{N}$ keys, where N is the number of regions. Thirdly, all nodes can communicate with each other in a group, which results in higher resiliency. Only three agents are required to ensure that all regions which are within Lee distance are connected. The resiliency of our scheme is much better than many state-of-the-art schemes. In our scheme we also use another measure

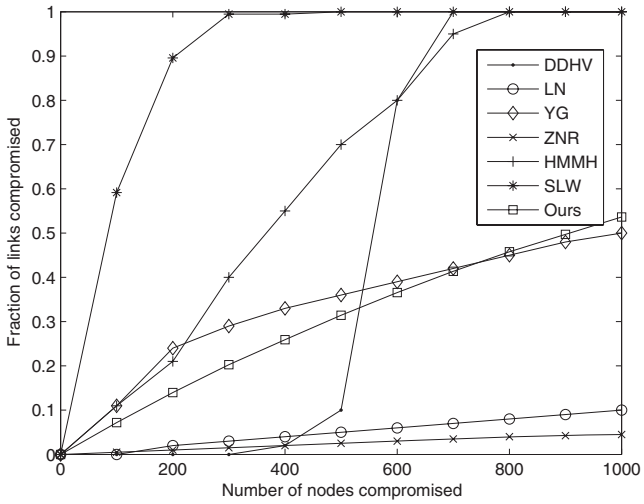


Fig. 4. Comparison of DDHV, LN, YG, ZNR, HMMH, SLW, and our scheme. (i)DDHV scheme has parameters $k = 200$, $\omega = 11$, and $\tau = 2$; (ii)LN scheme has parameters $k = 200$, $m = 60$, and $L = 1$; (iii)YG scheme has parameters $k = 100$; (iv)ZNR scheme has parameters $k = 100$; (v)HMMH scheme has parameters $k = 200$, $\omega = 27$, and $\tau = 3$; (vi)SLW scheme has parameters $k = 16$, $p = 11$, and $m = 4$; (vii)Our scheme has parameters $k = 12$. The size of the network in DDHV, LN, YG, ZNR, HMMH is 10000, for SLW it is 12100 and 16093 for our scheme.

of resiliency $V'(s)$ and $V''(s)$ (the fraction of nodes or regions disconnected when s nodes are compromised) which has not been considered in any deployment-knowledge-based network.

We present a comparison of resiliency of the several schemes with our scheme in Figure 4. We notice that our resiliency is better than most schemes. Also the number of keys in our scheme is surprisingly low compared to the other schemes.

6. CONCLUSION AND FUTURE RESEARCH

In this article we propose a new key predistribution scheme for a grid-group-based deployment scheme. In this scheme the adversarial region is divided into a number of squares. There are two types of sensor nodes of different power and storage capacity. Within a region all nodes can communicate with each other. When sensor nodes in different regions want to communicate, then this is done via special nodes called agents, which have more power and *mem*ory. We predistribute keys in nodes in the groups according to combinatorial structures called projective planes. We use another type of combinatorial designs called transversal deigns while predistributing keys in the agents. Only three agents are required for interregion communication, whatever the size of the network. All nodes within a group can communicate with each other. Also agents in one region can communicate with agents in other regions within the Lee sphere. Thus delays and errors due to multihop communication is reduced. Our scheme has the added advantage that very few keys have to be stored in the nodes compared to all existing schemes which use deployment knowledge. We measure

the resiliency of the network in terms of the fraction of links broken and get better results than existing schemes. Also we measure the resiliency in terms of the fraction of nodes or regions disconnected and get very good resiliency. The second parameter is important because a disconnected node or region cannot communicate at all. This had not been studied in earlier schemes which used deployment knowledge.

In our article we have approximated the communication region using the Lee sphere. This was done to simplify the calculations for resiliency. In the future we would like to do all our analysis considering the communication region to be circular.

REFERENCES

BLACKBURN, S. R., ETZION, T., MARTIN, K. M., AND PATERSON, M. B. 2008. Efficient key predistribution for grid-based wireless sensor networks. In *Proceedings of the International Conference on Information Theoretic Security (ICITS'08)*. R. Safavi-Naini, Ed. Lecture Notes in Computer Science, vol. 5155. Springer, 54–69.

BLOM, R. 1984. An optimal class of symmetric key generation systems. In *Proceedings of the Annual International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT'84)*. 335–338.

BLUNDO, C., SANTIS, A. D., HERZBERG, A., KUTTEN, S., VACCARO, U., AND YUNG, M. 1998. Perfectly secure key distribution for dynamic conferences. *Inf. Comput. 146*, 1, 1–23.

ÇAMTEPE, S. A. AND YENER, B. 2004. Combinatorial design of key distribution mechanisms for wireless sensor networks. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS'04)*. P. Samarati, P. Y. A. Ryan, D. Gollmann, and R. Molva, Eds. Lecture Notes in Computer Science, vol. 3193. Springer, 293–308.

ÇAMTEPE, S. A. AND YENER, B. 2005. Key distribution mechanisms for wiless sensor networks: A survey. Tech. rep. TR-05-07, Computer Science DePartment, Rensselaer Polytechnic Institute.

ÇAMTEPE, S. A. AND YENER, B. 2007. Combinatorial design of key distribution mechanisms for wireless sensor networks. *IEEE/ACM Trans. Netw. 15*, 2, 346–358.

ÇAMTEPE, S. A., YENER, B., AND YUNG, M. 2006. Expander graph based key distribution mechanisms in wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications*. 2262–2267.

CHAKRABARTI, D., MAITRA, S., AND ROY, B. K. 2006. A key pre-distribution scheme for wireless sensor networks: Merging blocks in combinatorial design. *Int. J. Inform. Sec. 5*, 2, 105–114.

CHAN, H. AND PERRIG, A. 2005. Pike: Peer intermediaries for key establishment in sensor networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom'05)*. IEEE, 524–535.

CHAN, H., PERRIG, A., AND SONG, D. X. 2003. Random key predistribution schemes for sensor networks. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society, 197–213.

CHENG, Y. AND AGRAWAL, D. P. 2007. An improved key distribution mechanism for large-scale hierarchical wireless sensor networks. *Ad Hoc Netw. 5*, 1, 35–48.

DAS, A. K. AND SENGUPTA, I. 2008. An effective group-based key establishment scheme for large-scale wireless sensor networks using bivariate polynomials. In *Proceedings of the International Conference on Communication System Software and Middleware (COMSWARE'08)*. IEEE, 9–16.

DU, W., DENG, J., HAN, Y. S., CHEN, S., AND VARSHNEY, P. K. 2004. A key management scheme for wireless sensor networks using deployment knowledge. In *Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom'04)*.

DU, W., DENG, J., HAN, Y. S., AND VARSHNEY, P. K. 2003. A pairwise key pre-distribution scheme for wireless sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*. ACM, 42–51.

DU, W., DENG, J., HAN, Y. S., AND VARSHNEY, P. K. 2006. A key predistribution scheme for sensor networks using deployment knowledge. *IEEE Trans. Depend. Sec. Comput. 3*, 1, 62–77.

DU, W., DENG, J., HAN, Y. S., VARSHNEY, P. K., KATZ, J., AND KHALILI, A. 2005. A pairwise key predistribution scheme for wireless sensor networks. *ACM Trans. Inform. Syst. Secur. 8*, 2, 228–258.

DU, X., XIAO, Y., GUIZANI, M., AND CHEN, H.-H. 2007. An effective key management scheme for heterogeneous sensor networks. *Ad Hoc Netw. 5*, 1, 24–34.

ESCHENAUER, L. AND GLIGOR, V. D. 2002. A key-management scheme for distributed sensor networks. In *Proceedings of the ACM Conference on Computer and Communications Security*. V. Atluri, Ed. ACM, 41–47.

FERREIRA, A. C., VILAÇA, M. A., OLIVEIRA, L. B., HABIB, E., WONG, H. C., AND LOUREIRO, A. A. F. 2005. On the security of cluster-based communication protocols for wireless sensor networks. In *Proceedings of the International Conference on Networking (ICN'05)*. P. Lorenz and P. Dini, Eds. Lecture Notes in Computer Science, vol. 3420. Springer, 449–458.

HEINZELMAN, W. R., CHANDRAKASAN, A., AND BALAKRISHNAN, H. 2000. Energy-Efficient communication protocol for wireless microsensor networks. In *Hawaii International Conference on System Sciences (HICSS'00)*.

HUANG, D. AND MEDHI, D. 2007. Secure pairwise key establishment in large-scale sensor networks: An area partitioning and multigroup key predistribution approach. *ACM Trans. Sensor Netw. 3*, 3.

HUANG, D., MEHTA, M., MEDHI, D., AND HARN, L. 2004. Location-Aware key management scheme for wireless sensor networks. In *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04)*. 29–42.

HUSSAIN, S., KAUSAR, F., AND MASOOD, A. 2007. An efficient key distribution scheme for heterogeneous sensor networks. In *Proceedings of the International Conference on Communications and Mobile Computing (IWCMC'07)*. M. Guizani, H.-H. Chen, and X. Zhang, Eds. ACM, 388–392.

HWANG, J. AND KIM, Y. 2004. Revisiting random key pre-distribution schemes for wireless sensor networks. In *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04)*. 43–52.

JAJODIA, S., ATLURI, V., AND JAEGER, T., EDS. 2003. *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*. ACM, New York.

JOLLY, G., KUSÇU, M. C., KOKATE, P., AND YOUNIS, M. F. 2003. A low-energy key management protocol for wireless sensor networks. In *Proceedings of the IEEE International Symposium on Computers and Communications (ISCC'03)*. IEEE Computer Society, 335–340.

LEE, J. AND STINSON, D. R. 2004. Deterministic key predistribution schemes for distributed sensor networks. In *Selected Areas in Cryptography*, H. Handschuh and M. A. Hasan, Eds. Lecture Notes in Computer Science, vol. 3357. Springer, 294–307.

LEE, J. AND STINSON, D. R. 2005. A combinatorial approach to key predistribution for distributed sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC'05)*.

LEE, J. AND STINSON, D. R. 2008. On the construction of practical key predistribution schemes çfor distributed sensor networks using combinatorial designs. *ACM Trans. Inform. Syst. Secur. 11*, 2.

LIU, D. AND NING, P. 2003. Establishing pairwise keys in distributed sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*. 52–61.

LIU, D. AND NING, P. 2005. Improving key predistribution with deployment knowledge in static sensor networks. *ACM Trans. Sensor Netw. 1*, 2, 204–239.

OLIVEIRA, L. B., FERREIRA, A. C., VILAÇA, M. A., WONG, H. C., BERN, M. W., DAHAB, R., AND LOUREIRO, A. A. F. 2007. Secleach - On the security of clustered sensor networks. *Signal Process. 87*, 12, 2882–2895.

OLIVEIRA, L. B., WONG, H. C., BERN, M. W., DAHAB, R., AND LOUREIRO, A. A. F. 2006. Secleach A random key distribution solution for securing clustered sensor networks. In *Proceedings of the IEEE International Symposium on Network Computing and Applications (NCA'06)*. IEEE Computer Society, 145–154.

PATERSON, M. B. AND STINSON, D. B. 2008. Two attacks on a sensor network key distribution scheme of Cheng and Agrawal. *J. Math. Crypt. 2*, 393–403.

PIETRO, R. D., MANCINI, L. V., AND MEI, A. 2006. Energy efficient node-to-node authentication and communication confidentiality in wireless sensor networks. *Wirel. Netw. 12*, 6, 709–721.

RUJ, S. 2009. Application of combinatorial structures to key predistribution in sensor networks and traitor tracing. Ph.D. thesis, Indian Statistical Institute, India.

RUJ, S., MAITRA, S., AND ROY, B. 2008. Key predistribution using transversal design on a grid of wireless sensor network. *Ad Hoc Sensor Wirel. Netw. 5*, 3-4, 247–264.

RUJ, S. AND ROY, B. K. 2007. Key predistribution using partially balanced designs in wireless sensor networks. In *Proceedings of the International Symposium on Image and Signal Processing and Analysis (ISPA'07)*. I. Stojmenovic, R. K. Thulasiram, L. T. Yang, W. Jia, M. Guo, and R. F. de Mello, Eds. Lecture Notes in Computer Science, vol. 4742. Springer, 431–445.

RUJ, S. AND ROY, B. K. 2008. Key establishment algorithms for some deterministic key predistribution schemes. In *Proceedings of the International Workshop on Security in Information Systems (WOSIS'08)*, A. Rodríguez, M. I. Y. del Valle, and E. Fernández-Medina, Eds. INSTICC Press, 68–77.

SIMONOVA, K., LING, A. C. H., AND WANG, X. S. 2006. Location-Aware key predistribution scheme for wide area wireless sensor networks. In *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'06)*. S. Zhu and D. Liu, Eds. ACM, 157–168.

STREET, A. P. AND STREET, D. J. 1987. *Combinatorics of Experimental Design*. Clarendon Press, Oxford, UK.

XIAO, Y., RAYI, V. K., SUN, B., DU, X., HU, F., AND GALLOWAY, M. 2007. A survey of key management schemes in wireless sensor networks. *Comput, Comm. 30*, 11-12, 2314–2341.

YOUNIS, M. F., GHUMMAN, K., AND ELTOWEISSY, M. 2006. Location-Aware combinatorial key management scheme for clustered sensor networks. *IEEE Trans. Parall. Distrib. Syst. 17,* 8, 865–882.

YU, Z. AND GUAN, Y. 2005. A key pre-distribution scheme using deployment knowledge for wireless sensor networks. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN'05)*. IEEE, 261–268.

YU, Z. AND GUAN, Y. 2008. A key management scheme using deployment knowledge for wireless sensor networks. *IEEE Trans. Parall. Distrib. Syst. 19*, 10, 1411–1425.

ZHOU, L., NI, J., AND RAVISHANKAR, C. V. 2006. Supporting secure communication and data collection in mobile sensor networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom'06)*. IEEE.

ZHU, S., SETIA, S., AND JAJODIA, S. 2003. Leap: Efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*. 62–72.

Received April 2008; revised March 2009; accepted March 2009