Simple, extensible and flexible random key predistribution schemes for wireless sensor networks using reusable key pools

Albert Levi · Sinan Emre Taşçı · Young Jae Lee ·
Yong Jae Lee · Ersoy Bayramoğlu · Murat Ergun

Abstract Sensor nodes are tiny, low-power, computationally limited and battery constrained electromechanical devices. A sensor node contains a sensing unit and a wireless communication unit. Sensor nodes are deployed over a field for sensing an event data in the environment and transfer it towards a base station over its wireless channel. In a typical application, vast amount of sensor nodes are deployed over a field which constitute a *sensor network*. Sensor nodes must be customized for a specific sensor network application before the deployment. This customization is needed not only for underlying networking application, but also for security related configurations. Random key predistribution mechanisms have been proposed to provide security for wireless sensor networks. In the literature, there are well known random key predistribution schemes. Some of these schemes are secure, but quite complex to apply in real-world applications due to their node-based customization requirements, while some other are easily applicable but they do not offer reasonable security. In this paper, we propose random key predistribution schemes for wireless sensor networks that provide varying ranges of security. The proposed schemes are easily applicable in real world scenarios due to their simplicity and relaxed node customization requirements. In this respect, our schemes provide a tradeoff. Moreover, our proposed schemes show a good extensibility property. We assume prior deployment knowledge. We examine performance of our schemes and compare them with well known random key predistribution schemes.

Keywords Security · Sensor network security · Key distribution · Sensor node customization · Resiliency

Introduction and related work

Wireless sensor networks (Akyildiz et al. 2002) have recently received remarkable attention. A sensor network contains a large number of tiny sensor nodes that sense data specific to that environment and report them to other nodes and to a base station (a.k.a. the *sink*) over a flexible infrastructure. Sensor networks can be used for different types of application scenarios such as military tracking, health care, environmental sensing and home automation (Akyildiz et al. 2002).

Sensor nodes are electromechanical devices. The basic manufacturing process of sensor nodes is not a customized one and usually yields generic electromechanical devices. In order to use sensor nodes in particular applications, they have to be configured, for example, by installing some software, and customized by loading some node-specific data. Especially security related cryptographic keys are the major source of customization. In the rest of this paper, we assume

This paper is an extended version of Tasci et al. (2008).

A. Levi (✉) · S. E. Taşçı · E. Bayramoğlu · M. Ergun
Sabancı University, Tuzla, Istanbul, Turkey
e-mail: levi@sabanciuniv.edu

S. E. Taşçı
e-mail: sinanemre@su.sabanciuniv.edu

M. Ergun
e-mail: mergun@su.sabanciuniv.edu

Y. J. Lee
Jeonju University, Jeonju, Korea
e-mail: leeyj@jj.ac.kr

Y. J. Lee
Tongmyong University, Busan, Korea
e-mail: leeyj@tu.ac.kr

E. Bayramoğlu
EPFL, Lausanne, Switzerland
e-mail: ersoy.bayramoglu@epfl.ch

that full manufacturing process of a sensor node includes configuration and customization on top of hardware manufacturing.

In some of these applications, sensor networks are deployed in hostile environments and over large geographical regions. When sensor networks are deployed in such hostile environments, security becomes a very important problem to be resolved. Sensor networks are subject to different types of security threats and attacks (Karlof and Wagner 2003). These include physical capture of a node, intentionally providing misleading information, impersonation, data modification, eavesdropping, etc.

Cryptography (Menezes et al. 1996) is an important tool to provide confidentiality and authentication type of security in data networks including sensor networks. In order to provide cryptographic security for sensor networks, first authentication and key management protocols must be applied among the sensor nodes and the sink. However, the architecture of sensor networks and limitations on sensor nodes do not allow well known protocols and cryptosystems to be used. Moreover, public key cryptography (Diffie and Hellman 1976; Rivest et al. 1978) based protocols are not suitable for sensor networks because of their computational and memory restrictions. For two party authentication, integrity and freshness, μTESLA scheme (Perrig et al. 2001), which is based on delayed key disclosure, is proposed. However, we still need pairwise keys to provide confidentiality. Actually, when pairwise cryptographic keys are distributed in a secure and effective manner, most of the security problems can be addressed via different protocols. Thus, we need effective key distribution mechanisms for sensor networks.

Eschenauer and Gligor (2002) proposed a random key predistribution mechanism for wireless sensor networks and led to an innovation in this area. In this mechanism, first a large global key pool is generated. Then, each node is loaded with a predefined number of keys that constitute its *key ring*. Keys of the key rings are picked from the global key pool in uniformly random fashion. All nodes are then deployed onto the field again in uniform random fashion. A securely communicating network can be formed with the key sharing information between sensor nodes. In other words, each node discovers whether it shares at least one key with its neighbors in pairwise manner; if this is the case, then a secure communication link is established between these node pairs. This scheme is called *basic scheme*. After the proposal of the basic scheme, some other random key predistribution schemes are proposed in the literature (Chan et al. 2003; Liu and Ning 2003; Du et al. 2003; Unlu et al. 2007).

The basic scheme did not consider any prior deployment knowledge and come up with an assumption that all nodes are deployed uniformly random on the deployment area. However, prior deployment knowledge may be utilized to improve the performance of a random key predistribution scheme. Although it may not be possible to previously know the exact deployed location of a node, it is possible to have an idea about approximate location of a node after deployment. Huang et al. (2004) and Du et al. (2004) proposed two such random key predistribution schemes that consider prior deployment knowledge.

The scheme by Du et al. (2004) is particularly important in this paper since our work is based on it. This scheme assumes a grid deployment mechanism. Nodes are assumed to be deployed in the center of each zone as a batch. Those batches of nodes are distributed over each zone according to Gaussian distribution, which is best fit the real world deployment scenarios. In this deployment model, the nodes in each batch are assumed to be close to each other. This is, actually, prior deployment knowledge exploited in Du et al. (2004). Keys are assigned to each node randomly by selecting from the key pool of the corresponding zone. Each zone shares keys with its neighbor zones. In this way, the nodes that are close to each other have a probability to share keys, but the distant nodes do not. However, key distribution mechanism of Du et al. (2004) is complicated and inconvenient since it requires various sensor node types for each zone. Key sharing computation is offline and the topology of the sensor network must be considered in this process. Therefore, the manufacturing process of the sensor nodes must be a customized one. As a payback to those problems, this scheme provides security and resiliency against node capture.

The sensor nodes are generally low-cost hardware devices with limited power, computational capacity and memory. The requirement of using vast amount of nodes in a particular sensor network deployment enforces the low-cost manufacturing process of the sensor nodes. Not only the hardware parts and their assembly, customization of the sensor nodes is also a cost factor. The scheme proposed in Du et al. (2004) performs well in terms of security and resiliency but it requires a customized set of sensor nodes per zone. Moreover, this set of nodes cannot be reused in another part of the network. On the other hand, the basic scheme (Eschenauer and Gligor 2002) does not perform as effective as Du et al. (2004), but it needs just one generic node type, so custom manufacturing is not needed. Our aim in this paper is to devise a tradeoff scheme between security/resiliency and ease of manufacturing via less customization.

In this paper, we propose two random key predistribution schemes that assume grid based deployment mechanism. The proposed mechanisms follow similar guidelines as the scheme proposed in Du et al. (2004). However, in our schemes the key pools of the zones are reused so that a simpler key distribution model is achieved with less distinct key pools. We do not aim to propose a unique solution to key distribution problem in sensor networks. Our motivation is to design simple and flexible key distribution schemes that are easily applicable, extensible and sufficiently secure for real world deployment and manufacturing scenarios. In this way, we provide a tradeoff scheme as compared to existing schemes in the literature.

The rest of this paper is organized as follows. The proposed schemes are explained in the next section. After that, comparative performance evaluation of the proposed schemes and the ones in the literature is given as another section. Last two sections summarize the discussions and the conclusions reached by this study.

Proposed schemes

We propose two specific distribution schemes, which assume prior deployment knowledge, for wireless sensor networks. The first scheme uses just two large key pools for the overall network and aims to achieve improved security as compared to the basic scheme. The second scheme is a bit more complicated and offers higher security as compared to the first one. This scheme uses two key pools for each line over the deployment grid.

In our schemes, similar to the scheme proposed in Du et al. (2004), batch of nodes are assumed to be distributed from a moving vehicle over several zones. Zone based distribution puts location knowledge and key sharing information together. In this way, the key ring size, which is the number of keys that a sensor node should store, is reduced.

The first scheme: ABAB

In this scheme, there exist two key pools A and B. These two key pools get their distinct keys from a global key pool S. Moreover, they share a common key pool, s, which is also picked from the global key pool S. In order to prepare the nodes for deployment, m (key ring size) keys are picked in uniformly random fashion from the key pool A or B according to the target deployment zone (generally in alternating manner). After that, nodes collected as batches and deployed onto each target zone. The motivation behind this is to design a simple key distribution scheme that is suitable for most of the sensor node deployment purposes. Actually the idea is to make use of that simple location knowledge while keeping the distribution as simple as possible. The ABAB scheme is depicted for a 2 × 2 zone in Fig. 1.

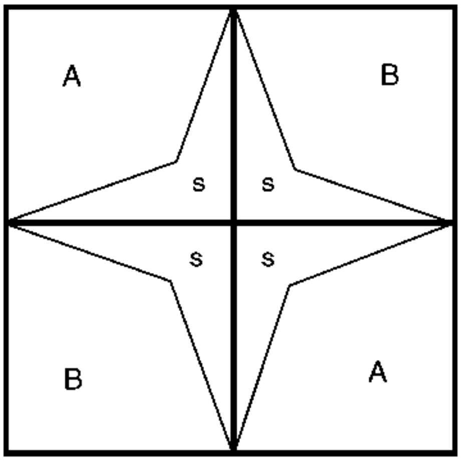

Fig. 1 Key pool selection of ABAB scheme

The amount of distinct keys and shared keys of key pools A and B are calculated as follows. Let a denote number of distinct keys in a key pool (A) or (B). Moreover, b denotes the size of the shared key pool s. The ratio of size of the shared key pool$s$ over the size of global key pool S is denoted as ω. In ABAB scheme, the size of the global key pool, $|S|$, is given as follows.

$$|S| = 2a + b \tag{1}$$

Since by definition $b = \omega \cdot |S|$, a is calculated as follows.

$$|S| = 2a + \omega \cdot |S| \Rightarrow a = \frac{(1-\omega)\cdot|S|}{2} \tag{2}$$

Moreover, the total number of keys in a particular key pool, which is denoted as K, is calculated as follows.

$$K = a + b = \frac{(1-\omega)\cdot|S|}{2} + \omega\cdot|S| = \frac{(1+\omega)\cdot|S|}{2} \tag{3}$$

Key pool generation in ABAB scheme is performed in a few steps as described below.

Step 1: Generate key pool s by picking b keys from the global key pool. Remove these keys from the global key pool.
Step 2: Generate a key pool A' by picking a keys of the remaining global key pool at random. Remove these keys from the global key pool.
Step 3: The remaining global key pool has a keys in it. Assign them to another key pool B'.
Step 4: Merge key pools s and A' in order to form key pool A.
Step 5: Merge key pools s and B' in order to form key pool B.

After the key pools A and B are generated, for each zone, m (key ring size) keys are randomly selected from key pool A or key pool B. These keys are stored in the nodes. Now, the nodes are ready to be deployed over the field. The deployment is generally in checkerboard manner such that side-by-side neighbors are of different key pools. A sample deployment is depicted in Fig. 2.

ABAB scheme shows a nice extensibility property. Whenever we need to add a new zone to the sensor network, depending on the neighboring key pool, either an A or a B zone can be added without rearranging the existing pools. Figure 3 shows and extension over the network shown in

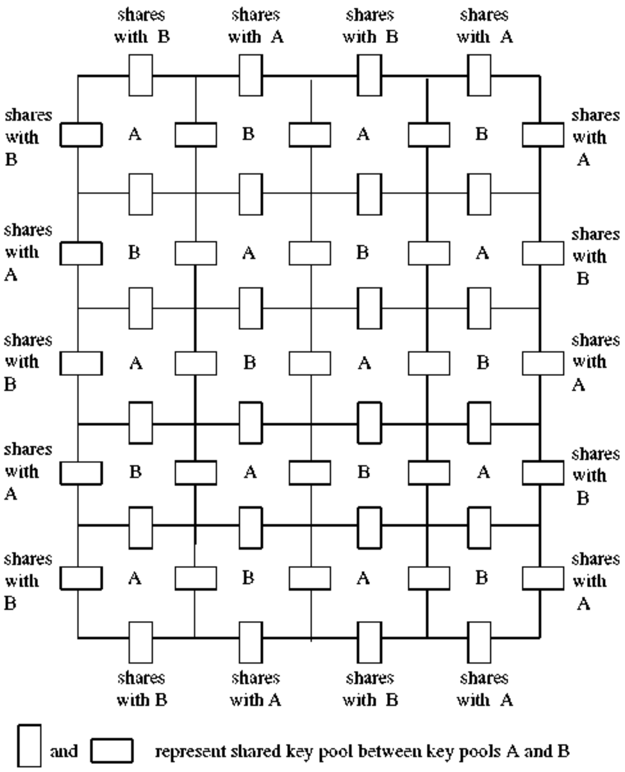

Fig. 2 A sample deployment in ABAB scheme

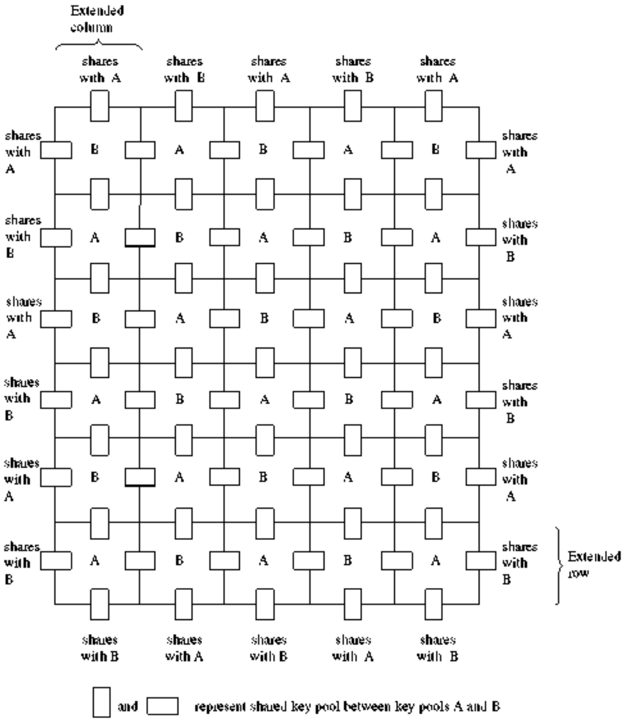

Fig. 3 A sample network extension in ABAB scheme

Fig. 2. As can be seen in Fig. 3, the network can be enlarged from, say, bottom and left by adding new *A* and *B* zones in alternating manner.

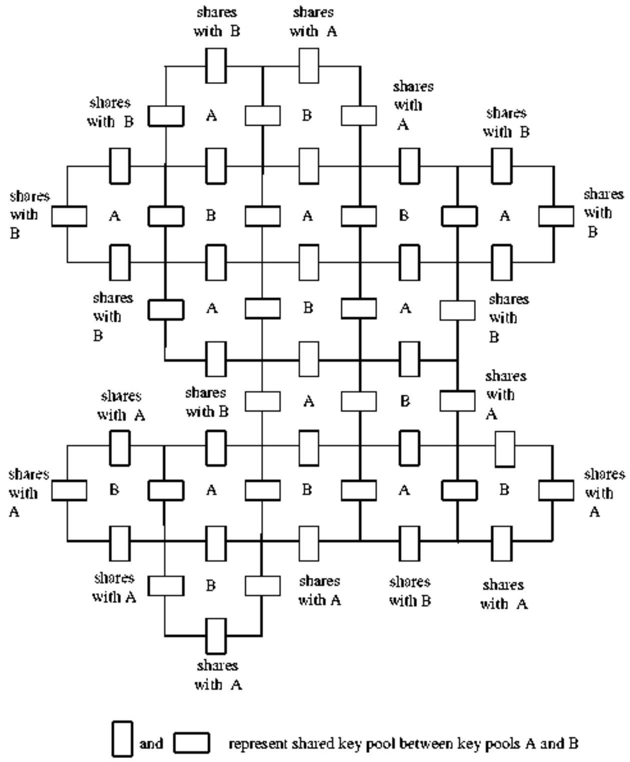

Fig. 4 A sample network covering an amorphous region in ABAB scheme

Moreover, ABAB scheme can be used to generate sensor fields of any topological or amorphous shape by arranging zones of *A* and *B* types in alternating manner. For example, Fig. 4 shows coverage for an amorphous area. It is worthwhile to mention here is that sensor network topology is not considered while creating nodes of key pools *A* and *B*. Thus, whatever the topology would be, previously manufactured nodes can be used in the deployment phase without needing to generate new sensor nodes via a customized manufacturing process. Needless to say that the extensibility property still holds for this type of networks as well.

The second scheme: ABCD

ABAB scheme is easily applicable in sensor networks but it has a resiliency problem since same keys are used in different zones several times. Capture of a node causes compromise of keys that are used in other zones (see "Performance evaluation" section for details). In order to solve this problem, we propose another scheme, called ABCD scheme.

Decreasing the number of keys of the key pool of each zone is a way of increasing the resiliency. To do so, number of key pools must be increased up to a certain limit if the global key pool size is fixed. However, recurrence of key pools in different zones must still be provided for the sake of simplicity and applicability. Under these considerations,

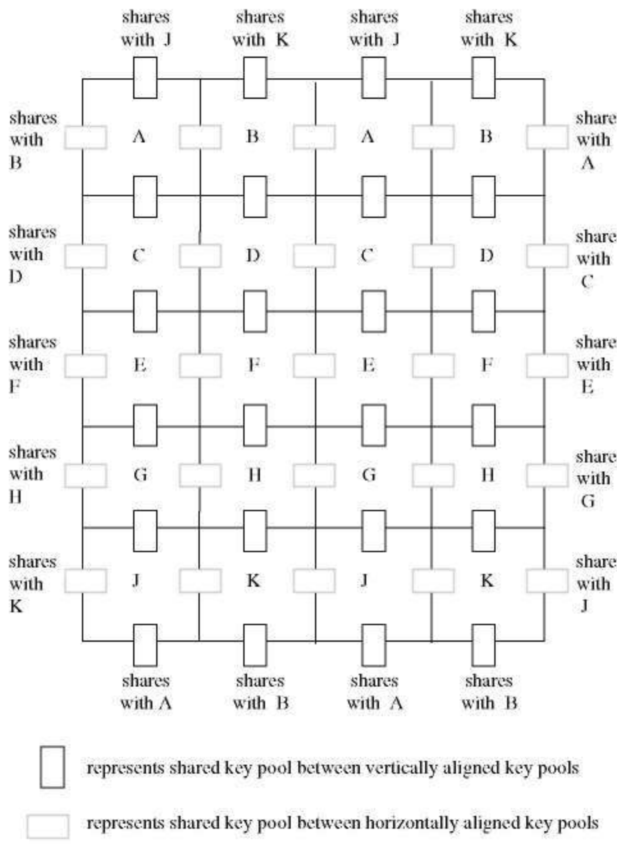

Fig. 5 A sample deployment in ABCD scheme

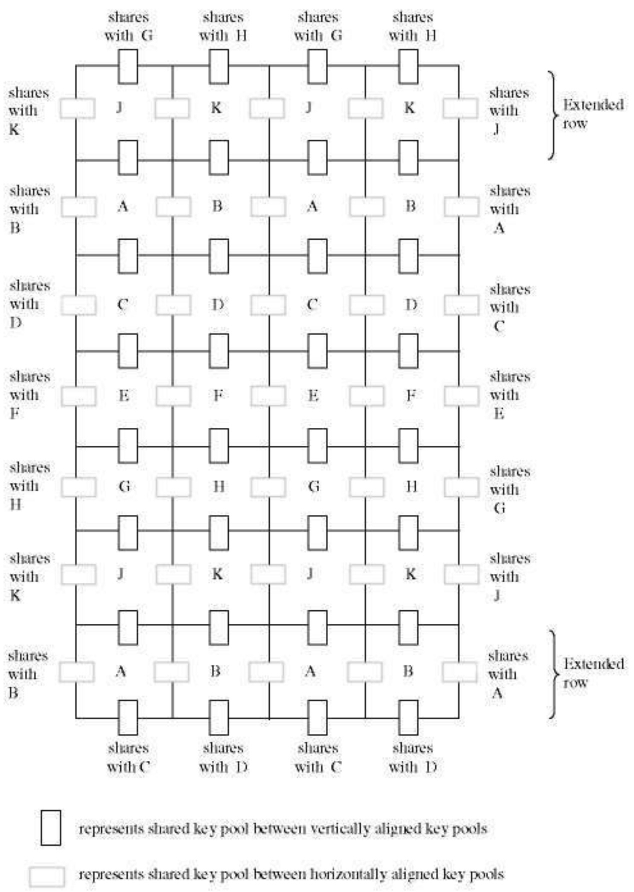

Fig. 6 Vertical extension in ABCD scheme

a new scheme, called ABCD scheme, is designed. In ABCD scheme, two different key pools are generated for each line of deployment. These two key pools share some number of keys with its neighbors both vertically and horizontally. For instance, assume that key pools A and B are generated for the first line of deployment. Pools A and B share some number of keys with each other. Key pools C and D that are generated for the second line of deployment share the same number of keys. Moreover, key pool C shares keys with key pool A and key pool D shares keys with key pool B as well. Moreover, the first and the last lines also share keys vertically. After generation of all key pools, zones are deployed in alternating manner as depicted in Fig. 5.

Number of lines with different key pool pairs is a system parameter. However, this does not mean that the network cannot enlarge after the initial deployment. Horizontal enlargement is possible by deploying zones that use the key pools of particular lines in alternating manner. Vertical enlargement from bottom is possible by deploying lines that reuse the same key pools starting from the first line. Similarly, vertical enlargement from top is possible by deploying lines that reuse the same key pools from backwards starting from the last line. Such connections are possible since the key pools of the last and the first lines share keys vertically. Figure 6 shows vertical enlargement from both top and bottom. Moreover, as in the ABAB scheme, ABCD scheme can be used to cover fields of any shape.

ABCD scheme has the same idea of ABAB scheme, which is reuse of the same key pools in different zones. However, ABCD scheme aims to come up with a more efficient and resilient scheme by using more key pools.

In this scheme, there is a tradeoff between the deployment simplicity, and local connectivity and resiliency. Increasing the key pool size for a zone (i.e. decreasing the number of key pools) makes the deployment simple, but the resiliency and connectivity of the system reduce. In ABCD scheme, it is possible to increase the number of key pools in a controlled way so that the resiliency of the system can be kept in required levels.

In ABCD scheme, there are $2r$ different key pools, where r is the number of rows of deployment (r is taken as 5 in Fig. 5). These key pools are denoted as $P_{i,j}$, where $i = 1, \ldots, r$ and $j = 1, 2$. Each key pool shares same amount of keys, denoted as b, with upper, lower and left/right neighbors. The ratio of each key pool share, b, over the total number of keys in a pool is denoted as ψ . Let a denote the number of keys, which are not shared, of a particular pool. Moreover, S denotes the global key pool and $|S|$ is the number of keys in S. Under these considerations, the total number of keys in a particular key pool, which is denoted as K, is calculated as follows.

$$K = a + 3b \tag{4}$$

Since $b = \psi K$, b is calculated as follows.

$$b = \psi(a + 3b) = \psi a + 3\psi b \Rightarrow b = \frac{\psi}{1 - 3\psi} a \tag{5}$$

Moreover, the ABCD scheme dictates that the keys in the global key pool S are to be distributed according to following equation.

$$|S| = 2ra + 3rb \tag{6}$$

When we substitute Eq. 5 in 6, we end up with the formula for a as follows.

$$|S| = 2ra + 3ra\frac{\psi}{1 - 3\psi} \Rightarrow a = \frac{|S|}{2r + 3r\frac{\psi}{1-3\psi}} \tag{7}$$

Having calculated the values of a and b for a particular sensor network using the system parameters r, ψ and $|S|$, horizontal and vertical arrangement of key pools are simply done as follows.

Step 1: For key pool $P_{i,1}$, $i = 1, 2, \ldots, r$, select a keys from the global key pool S. Then, remove those keys from S and assign them to$P_{i,1}$. Similarly, for key pool $P_{i,2}$, $i = 1, 2, \ldots, r$, select another set of a keys from the global key pool S. Then, remove those keys from S and assign them to$P_{i,2}$. For key pools $P_{i,1}$ and $P_{i,2}$, select b keys from S. Then, remove those keys from S and assign them to both $P_{i,1}$ and $P_{i,2}$.

Step 2: For groups $P_{i,j}$ and $P_{(i\,\mathrm{mod}\,r)+1,j}$, $i = 1, \ldots, r$ and $j = 1, 2$, select b keys from S. Then remove them from S and assign those keys to both $P_{i,j}$ and $P_{(i\,\mathrm{mod}\,r)+1,j}$.

All the key pools for all zones are now created. After randomly picking m (key ring size) keys for each node from the corresponding key pool, nodes are ready for deployment.

Direct key and path key establishment

In previous subsections, we describe how the key pools and, consequently, the key rings of individual sensor nodes are generated in our schemes. After the deployment, in order to establish pairwise keys, the sensor nodes must go through two more phases: (i) direct key establishment phase and (ii) path key establishment phase. These phases are the same as the basic scheme (Eschenauer and Gligor 2002) and the scheme proposed in Du et al. (2004). In direct key establishment phase, each node queries each of its neighbors to understand if it shares at least one key; if it shares, then a secure communication link is established between these two neighboring nodes. In path key establishment phase, neighboring nodes that do not directly share keys try to establish secure link over their securely connected neighbors via some hops.

Performance evaluation

In this section, an analysis for the performance of ABAB and ABCD schemes will be given, and compared to basic scheme (Eschenauer and Gligor 2002) and Du et al. (2004) scheme. Performance evaluation is performed via simulations using MATLAB.

Performance metrics

We analyze four different performance metrics: *local connectivity*, *global connectivity*, *resiliency* and *communication cost*. These metrics are used in basic scheme and Du et al.'s scheme as well. In this subsection, these metrics are described.

Local connectivity is the probability that two neighboring nodes share a key. This is possible when there is at least one common key in the key rings of these neighboring nodes. Higher local connectivity means more secure links established after the direct key establishment phase. Thus, it is preferable to have high local connectivity in a system. In simulations, for each node, x, we count total number of neighbors in the communication range of x, and the number of neighbors that x shares a key. Using these values, local connectivity, p, is estimated as below.

$$p = \frac{\sum_{i=1}^{N} \frac{\lambda_i}{\gamma_i}}{N} \tag{8}$$

where N is the number of nodes in the network, λ_i is the number of neighbors that node i shares at least one key, and γ_i is the number of nodes in the communication range of node i.

After the direct key establishment phase, the nodes that share a key create a graph, G, that may have more than one connected components. *Global connectivity* is defined as $|G_s|/|G|$ where G_s refers to the largest isolated component of G. Nodes that cannot communicate with any other node are excluded from G because this is caused by the deployment distribution.

In the path key establishment phase, some nodes share key over a trusted path. The number of hops in these paths is an important performance metric called the *communication cost*.

It is assumed that whenever a sensor node is captured by an attacker, all the keys stored in this node are automatically compromised. These compromised keys may be in use to secure some other links in another part of the network. Therefore, capture of a node does not only affect the captured node, but also some other nodes. The ratio of all compromised communication links between uncaptured nodes over all secure communication links gives the *fraction of the communications compromised*. Resiliency is the fraction of remaining secure communication links, but in our tests we use the fraction of communications compromised, which is 1-*resiliency*.

There is a tradeoff between local connectivity and resiliency. In order increase the local connectivity in a particular scheme, the key ring size (number of keys in each node) should be increased. This, in turn, causes lower resiliency since more keys are compromised when a node is captured.

Parameters and simulation methodology

The deployment area is a 10 × 10 grid and each zone in this grid is 100 m × 100 m area. The global key pool size, $|S|$, is 100000.

For the ABAB scheme, ω is taken as 0.1. For the ABCD scheme, r is taken as 10, and ψ is taken as 0.1. The reasons of selection ω and ψ parameters are explained in the next subsection.

Our extensive analyses show that our ABAB scheme performs comparable to the basic scheme (Eschenauer and Gligor 2002). Since basic scheme assumes uniformly random distribution of nodes, in ABAB simulations we also assumed uniform random distribution for the sake of fairness of comparison.

In our ABCD scheme, for each zone, nodes are assumed to be deployed airborne as batches from a moving vehicle such as an airplane. These batches are deployed targeting the center of each zone. However, deployed nodes are diversified from the center of that zone according to two-dimensional Gaussian distribution where $2\sigma = 100$ m. Communication range of a sensor node is 40m. Here one should note that the scheme by Du et al. (2004) also assumes the same deployment model and parameters. These model and parameters are selected to be compatible with this scheme.

Selection criteria for key sharing ratios

In both ABAB and ABCD schemes, we employ some parameters in order to decide on the size of the key pools. Other than the global key pool size and the number of key pools (for ABCD only), the most important parameter is the fraction of shared key pools. In ABAB, this parameter is denoted as ω, which means the ratio of the shared key pool size over the global key pool size. In ABCD, this parameter, which is denoted as ψ, is the ratio of the shared key pool size over the total number of keys in a zone key pool. We have performed tests separately for both ABAB and ABCD schemes in order to decide on the best values for these two parameters.

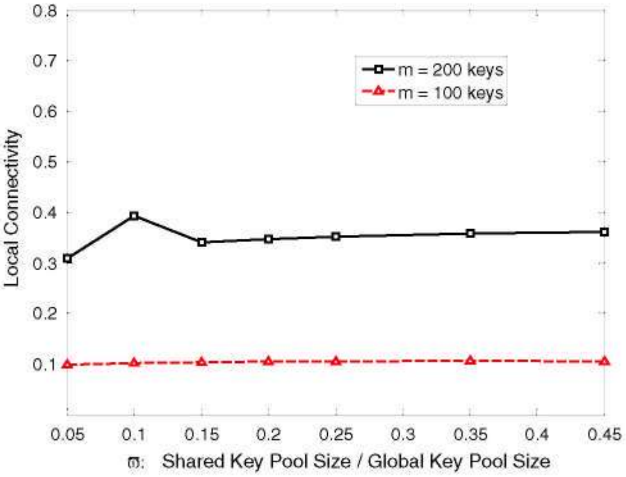

Fig. 7 Deciding simulation parameters for ABAB scheme

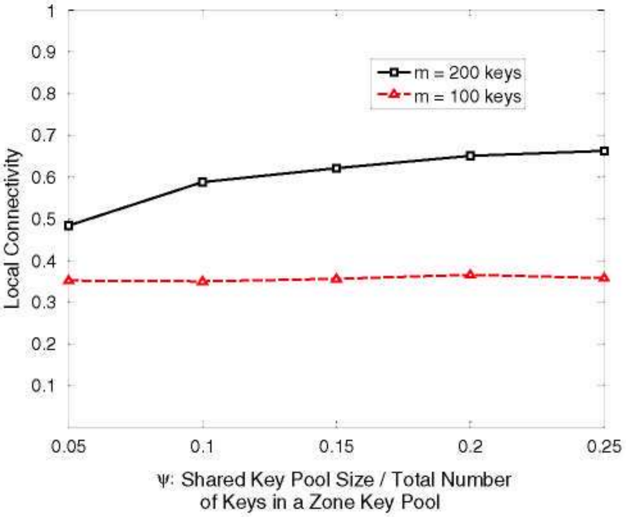

Fig. 8 Deciding simulation parameters for ABCD scheme

In Fig. 7, local connectivity of ABAB scheme is simulated with different ω values and with two different key ring sizes, $m = 100$ and $m = 200$. As can be seen from this figure, when $\omega = 0.1$ and $m = 200$, local connectivity increases and for remaining ω values, local connectivity is almost same. For the case where $m = 100$, local connectivity curve is almost flat which means that for small key ring sizes, the effect of ω is not significant. Based on these observations, for the sake of keeping the network more connected, ω value is taken as 0.1 for all ABAB simulations.

We performed similar tests to determine the ψ value for ABCD scheme. In Fig. 8, local connectivity of ABCD scheme is simulated with different ψ values. Again two different key ring sizes are used: $m = 100$ and $m = 200$. For the case of $m = 100$, local connectivity does not change with ψ. Therefore, as in ABAB case, for this m value ψ is not significant. On the other hand, when $m = 200$, local connectivity

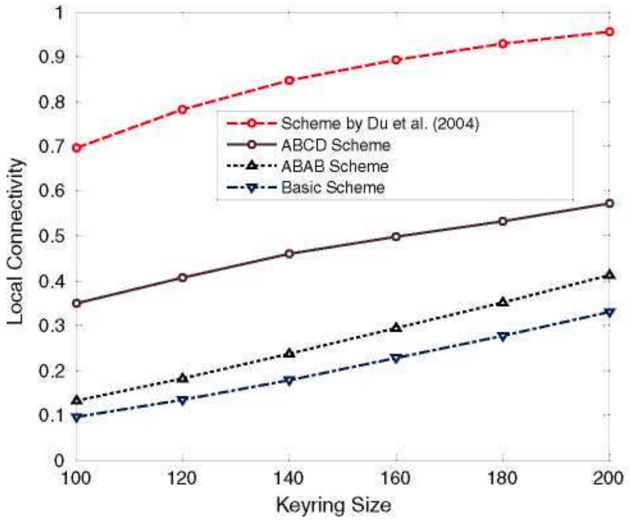

Fig. 9 Key ring size versus local connectivity for four different schemes

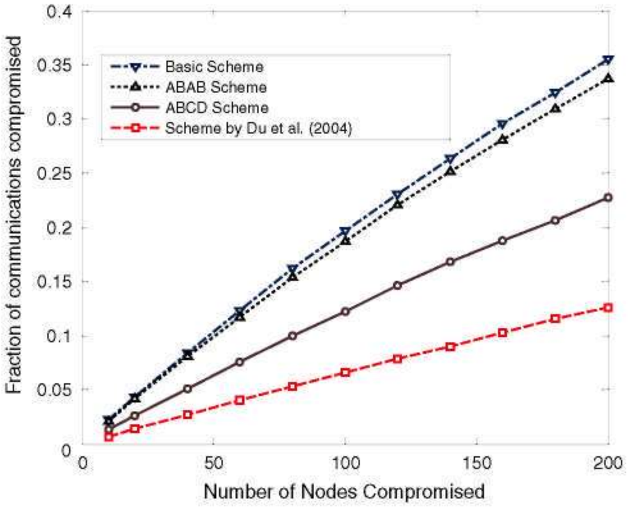

Fig. 10 Comparison of 1-resiliency performance versus number of nodes captured of four schemes

increases as ψ value increases. Although this increase is continuous, the rate of increase reduces after $\psi = 0.1$. Increase in local connectivity continues until $\psi = 0.25$, but ψ value is selected as 0.1. The reason of this selection is that since the size of the key pools is smaller in ABCD scheme as compared to ABAB scheme, same keys will be used frequently which negatively affects the resiliency of the network.

Simulation results

In Fig. 9, local connectivity performances of basic (Eschenauer and Gligor 2002), Du et al. (2004), proposed ABAB and ABCD schemes are presented versus different key ring sizes. As expected, local connectivity increases as the number of keys stored in a node (i.e. the key ring size) increases in all four schemes. The highest local connectivity is provided by Du et al. scheme. The main reason behind this performance is its customized (and also complicated) design of all key pools. For each zone, a new key pool and therefore a new set of sensor nodes need to be manufactured. ABCD scheme performs well as compared to the basic scheme since again it makes use of deployment knowledge. Although less than Du et al. scheme, the local connectivity of ABCD scheme is in acceptable level for a practical sensor network application. Moreover, ABCD scheme is less complicated than Du et al. scheme in terms of key pool generation and customization in sensor node manufacturing since the same key pools can be reused several times in the network. For example, in Du et al. (2004) scheme 100 different key pools must be generated and therefore 100 different sets of sensor nodes must be manufactured. However in ABCD scheme, only 20 different pools and sensor node sets are needed. In the ABAB scheme, this value is just 2. However, as a tradeoff of this simplicity, its local connectivity is lower than Du et al. scheme and ABCD schemes. Nevertheless, the local connectivity of the ABAB scheme is still better than the basic scheme.

Figure 10 depicts 1-*resiliency* performance (i.e. *fraction of communication links compromised*) of all four schemes versus varying number of nodes captured. In order to make fair comparison, same local connectivity, 33%, is assumed in all schemes. That corresponds to key ring sizes of 200 keys for the basic scheme (Eschenauer and Gligor 2002); 172 keys for ABAB; 95 keys for ABCD and 46 keys for Du et al. (2004) scheme. Actually, these key ring sizes determine the resiliency performance; as mentioned before, less number of keys in the key ring means a more resilient system. As a matter of fact, according to the simulation results, the scheme proposed in Du et al. (2004) offers the best resiliency against the node capture. This is due to the fact that this scheme decreases the number of keys to be used for each node. In other words, nodes that are deployed on the same zone need to have less number of keys to provide the same connectivity as compared to other schemes. ABCD scheme has also a substantial decrease in the number of keys to be deployed in each node. Even it is not as resilient as the scheme in Du et al. (2004), ABCD scheme provides better resiliency than both the basic scheme and ABAB scheme. This resiliency of ABCD scheme is applicable when real world deployment scenarios are considered. The resiliency of ABAB is better than the basic scheme, but not as good as ABCD and Du et al. schemes.

Global connectivity is another important metric of the key predistribution schemes as explained in Performance Metrics Subsection. Our comparative simulation results depicted in Fig. 11 show that the scheme by Du et al. (2004) is slightly

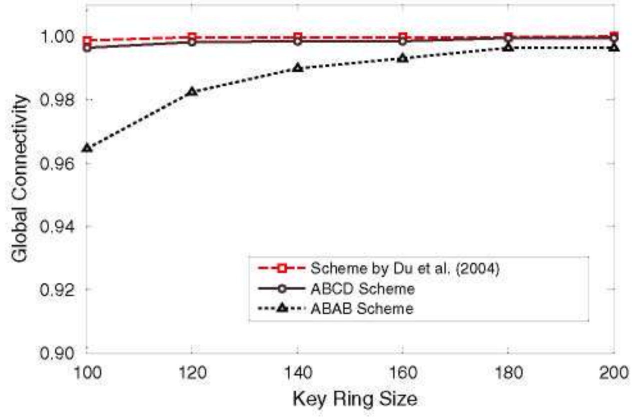

Fig. 11 Global connectivity comparison

better than ABCD scheme. The global connectivity provided by ABAB scheme with 150–200 keys is also sufficient for a sensor network.

However, making use of global connectivity is not free. In order to establish secure links among the neighbors, the nodes may need to involve in path key establishment process via secure routes with different number of hops. Establishment of a secure link via one hop corresponds to direct key establishment phase. In other words, the fraction of links that are established via one hop is the same as local connectivity. Establishment of secure links via more than one hops require flooding, thus it is not so practical to go beyond three hops since the number of packets in the network dramatically increase after that. Figures 12 and 13 show fraction of secure links established in one, two and three hops for different key ring sizes in ABAB and ABCD schemes, respectively. This is actually the *communication cost* metric described in Performance Metrics Subsection. As can be seen from Fig. 12, in ABAB scheme almost all of the possible secure links are established via at most three hops when the key ring size is greater than or equal to 150. Moreover, most of the links are established in one or two hops. The performance of ABCD scheme is even better. As shown in Fig. 13, for key ring sizes greater than or equal to 100, three hops become sufficient to generate all secure links. More than 90% of the secure links are established in one or two hops when the key ring size is at least 150.

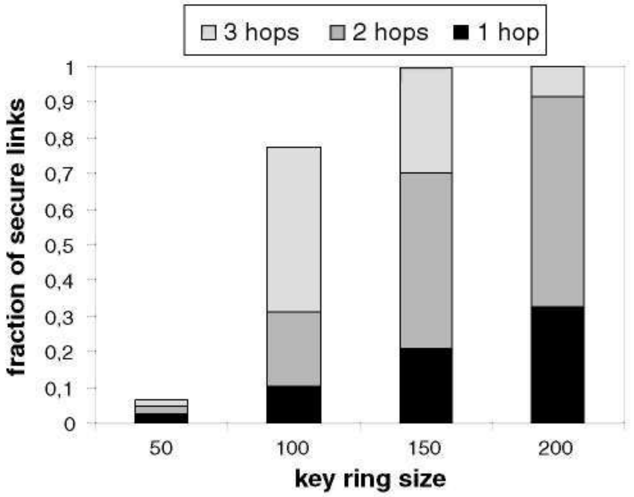

Fig. 12 Analysis of number of hops needed to establish secure links in ABAB scheme

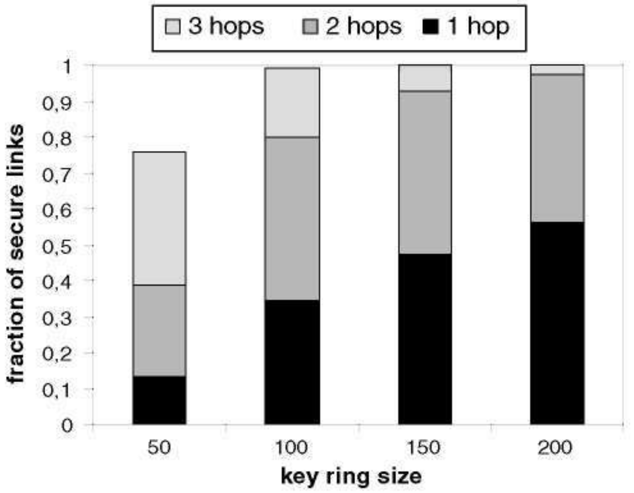

Fig. 13 Analysis of number of hops needed to establish secure links in ABCD scheme

Effect of gaussian distribution in ABAB scheme

In the above analyses, we assumed that the nodes are distributed uniformly random in ABAB scheme in order to have a fair comparison with the basic scheme (Eschenauer and Gligor 2002). However, if we adopt a zone-based airborne deployment model as in the case of Du et al. (2004) scheme, the distribution of the nodes must be 2-dimensional Gaussian. In this section, we analyze the *local connectivity* and 1-*resiliency* performance (i.e. *fraction of communication links compromised*) performances of the ABAB scheme with the same parameters as before but the nodes are distributed using 2-dimensional Gaussian distribution. An important issue about this distribution is its standard deviation, σ. In the airborne deployment scenario, standard deviation can be decreased by deploying nodes from a lower altitude. Similarly, standard deviation increases when the deployment altitude is higher. We run our simulations with four different σ values: 25, 30, 40 and 50. The local connectivity for these cases is depicted in Fig. 14. As can be seen from this figure, local connectivity increases as the standard deviation decreases. The reason of this behavior is that when standard deviation is decreased, the nodes do not get dispersed after the deployment and tend to remain in their home zones. It is

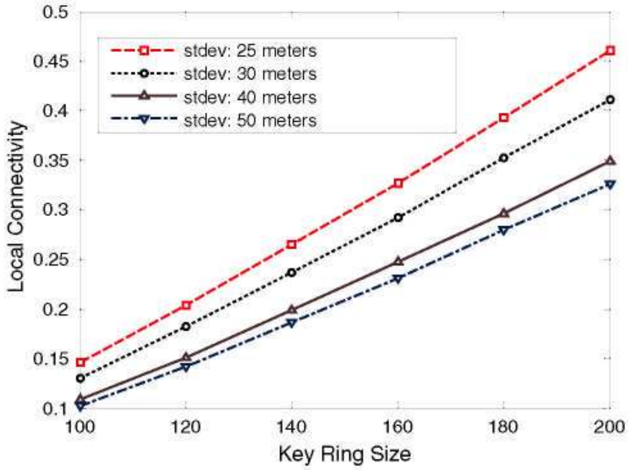

Fig. 14 Local connectivity of ABAB scheme with different standard deviation values for the Gaussian distribution

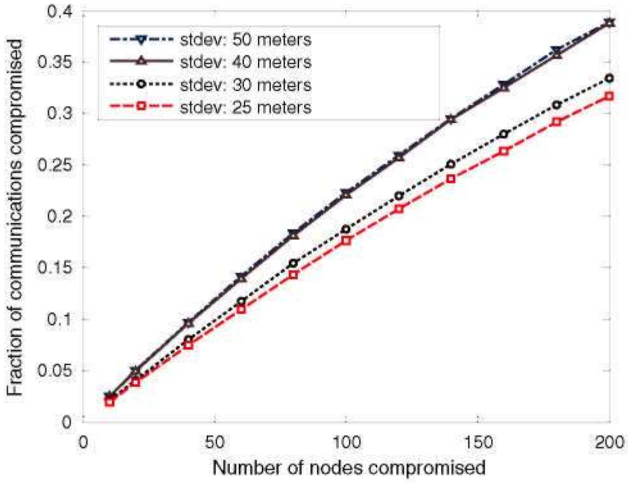

Fig. 15 1-resiliency performance of ABAB scheme with different standard deviation values for the Gaussian distribution

more probable to share a key between two nodes of the same zone as compared to the nodes that belong to different zones.

The *fraction of communication links compromised* performance of the ABAB scheme with the same standard deviation values is given in Fig. 15. As can be seen in the figure, reduced standard deviation has a positive affect in resiliency as well. In other words, less links are compromised as the standard deviation decreases. Actually, it is quite counter-intuitive to have improved connectivity with improved resiliency since these two metrics are known to be inversely related. This inverse relation is valid for the cases where local connectivity increases due to increased key ring size. However, in reduced standard deviation cases, local connectivity increases by keeping the nodes close to each other in their home zones. In this way, fewer keys of the shared key pools are used to establish secure links that causes a decrease in fraction of additional links compromised.

Table 1 The number of key pools needed to cover a sensor field of n-by-n zones for different schemes

Key distribution Scheme	Number of key pools
ABAB (proposed scheme)	2
ABCD (proposed scheme)	$2n$
Du et al. (2004) scheme	n^2

Discussions

Our ABAB and ABCD schemes have some unique features. These features will be detailed in this section.

The key pools of ABAB and ABCD are periodically reusable such that a particular key pool can be used to cover different zones on the field. In this way, same area can be covered using fewer amounts of key pools as compared to Du et al. (2004) scheme. In ABAB, only two key pools are used for the entire network independent of the area of the deployment field. In ABCD scheme, two key pools are used per deployment line. However, in Du et al. scheme the number of key pools is the same as the number of zones in the field. Table 1 shows the number of key pools needed to cover a square field that contains n × n zones. For example, in order to cover a field with 10 × 10 zones, ABCD scheme uses 20 key pools, while Du et al. scheme uses 100 key pools.

The cost of manufacturing of sensor nodes increases as the customization requirements increase. Not only during manufacturing, but also during supply-chain operations different types of nodes increase operational costs. As can be seen from Table 1, the number of key pools is fixed in our ABAB scheme. The increase in the number of key pools is linear with respect to the size of field in our proposed ABCD scheme. However, in Du et al. (2004) scheme the increase in number of key pools is quadratic; hence this scheme incurs more customization and operational costs as compared to the proposed schemes.

Moreover, in both ABAB and ABCD schemes, the key pools are generated as generic key pools such that the shape (including amorphous shapes) and the size of the sensor field are not production-level parameters. In other words, the sensor nodes can be manufactured independent of the deployment area. In this way, we avoid order-based manufacturing. However, in Du et al. scheme the key pool generations are based on the size and the shape of the field.

Whenever a key pool is generated in both ABAB and ABCD schemes, its four neighbors at four sides are explicitly defined including the ones at the boundaries of the field. Moreover, the deployment is periodic in both schemes such that the deployment patterns eventually repeat. These facts provide us an important advantage in *extensibility*. After the deployment, whenever a new zone is to be added to the network, the existing key pool arrangement signifies the zone type to be added. In this way, the sensor field can be indefinitely extended at any direction by reusing the existing key pools. This is valid for both ABAB and ABCD schemes. However, Du et al. scheme is designed for a fixed area and cannot be extended after the initial deployment.

However, the advantages mentioned above do not come for free. Although there is not a significant difference between ABCD scheme and Du et al. scheme in terms of *global connectivity*, ABCD scheme performs worse in *local connectivity* and *resiliency* measures as compared to Du et al. scheme. This performance difference is due to the fact that the global key pool is divided into more subpools in Du et al. scheme. In this way, with smaller number of keys per node, the probability of key sharing increases. In parallel to this fact, using less keys per node means increased resiliency.

In order to increase the performance of our scheme, it is possible to increase the number of key pools by sacrificing from the manufacturing/operational cost advantage. As an extreme case, our ABCD scheme can be modified to have the same amount of key pools as Du et al. scheme. According to our analysis of this extreme case, 46 keys per node are sufficient in order to obtain 33% of local connectivity. This value (46 keys) is also the same in Du et al. scheme for the same level of local connectivity. Moreover, the resiliency behaviors are also similar. In this extreme case scenario, although we sacrifice from the manufacturing and operational cost advantage, the extensibility advantage still exists since the key pools at the boundaries of the area share keys with other pools and these pools can be reused in case of an extension.

Conclusions

This paper presents two random key predistribution schemes, ABAB and ABCD, based on basic model (Eschenauer and Gligor 2002) and Du et al. (2004) scheme. Our models assume limited prior deployment knowledge. The aim of our work is making a tradeoff between security, and simplicity in node manufacturing, key distribution, and deployment. Our schemes use less key pools that can be reused for the nodes deployed in various zones. In this way, customization requirements of the production are relaxed and we achieve a flexible deployment model. Moreover, our schemes show a good extensibility property such that the sensor network field can easily be extended in an unplanned manner without making prior arrangements.

Our analyses show that our schemes perform better than Eschenauer and Gligor (2002), but worse than Du et al. (2004) in terms of connectivity and resiliency. However, by increasing the number of key pools of our ABCD scheme, system performance could be improved. Thus, it is possible to end up with higher performance by sacrificing from customization benefit (but not from extensibility). Our models offer a tradeoff in this manner.

Acknowledgments This work is supported by Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant 104E071.

References

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine, 40(8)*, 102–114. doi:10.1109/MCOM.2002.1024422.

Chan, H., Perrig, A., & Song, D. (2003). Random key predistribution schemes for sensor networks. In *Proceedings of IEEE symposium on security and privacy* (pp. 197–213).

Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory, 22*, 644–654. doi:10.1109/TIT.1976.1055638.

Du, W., Deng, J., Han, Y. S., Chen, S., & Varshney, P. K. (2004). A key management scheme for wireless sensor networks using deployment knowledge. In *Proceedings of IEEE Infocom*.

Du, W., Deng, J., Han, Y. S., & Varshney, P. K. (2003). A pairwise key pre-distribution scheme for wireless sensor networks. In *Proceedings of 10th ACM conference on computer and communications security* (pp. 42–51).

Eschenauer, L., & Gligor, V. D. (2002). A key-management scheme for distributed sensor networks. In *Proceedings of 9th ACM conference on computer and communications security* (pp. 41–47).

Huang, D., Mehta, M., Medhi, D., & 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* (pp. 29–42).

Karlof, C., & Wagner, D. (2003). Secure routing in wireless sensor networks: Attacks and countermeasures. In *Proceedings of first IEEE international workshop on sensor network protocols and applications*.

Liu, D., & Ning, P. (2003). Establishing pairwise keys in distributed sensor networks. In *Proceedings of 10th ACM conference on computer and communications security* (pp. 52–61).

Menezes, A. J., Vanstone, V. A., & Van Oorschot, P. C. (1996). *Handbook of Applied Cryptography*. CRC Press.

Perrig, A., Szewczyk, R., Wen, V., Cullar, D., & Tygar, J. D. (2001). Spins: Security protocols for sensor networks. In *Proceedings of 7th annual ACM/IEEE international conference on mobile computing and networking (MobiCom)* (pp. 189–199).

Rivest, R. L., Shamir, A., & Adleman, L. M. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM, 21*(2), 120–126. doi:10.1145/359340.359342.

Tasci, S. E., Bayramoglu, E., & Levi, A. (2008). Simple and flexible random key predistribution schemes for wireless sensor networks using deployment knowledge. In *Proceedings of 2nd international conference on information security and assurance (ISA 2008)* (pp. 488–494). IEEE Computer Society.

Unlu, A., Armagan, O., Levi, A., Savas, E., & Ercetin, O. (2007). Key predistribution schemes for sensor networks for continuous deployment scenario. In *Proceedings of IFIP/TC6 networking 2007*. LNCS (Vol. 4479, pp. 239–250). Springer.