Two-Tier, Scalable and Highly Resilient Key Predistribution Scheme for Location-Aware Wireless Sensor Network Deployments

Abdülhakim Ünlü · Albert Levi

Abstract We propose a probabilistic key predistribution scheme for wireless sensor networks, where keying materials are distributed to sensor nodes for secure communication. We use a two-tier approach in which there are two types of nodes: regular nodes and agent nodes. Agent nodes are more capable than regular nodes. Our node deployment model is zone-based such that the nodes that may end up with closer positions on ground are grouped together. The keying material of nodes that belong to different zones is non-overlapping. However, it is still possible for nodes that belong to different zones to communicate with each other via agent nodes when needed. We give a comparative analysis of our scheme through simulations and show that our scheme provides good connectivity figures at reasonable communication cost by using minimal flooding in key distribution. Moreover, we show that our scheme is scalable such that no extra overhead in incurred in case of increased number of nodes and sensor field size. Most importantly, simulation results show that our scheme is highly resilient to node captures.

Keywords security · authentication · key management · sensor networks · resiliency against node capture attacks

This paper is extended version of [19].

A. Ünlü · A. Levi (✉)
Faculty of Engineering and Natural Sciences,
Sabancı University, Orhanlı,
Tuzla, Istanbul 34956, Turkey
e-mail: levi@sabanciuniv.edu

A. Ünlü
e-mail: aunlu@su.sabanciuniv.edu

1 Introduction

When sensor networks [3] are used in a hostile setting, confidentiality, confidentiality and authenticity of communication among the sensor nodes should be provided. While fulfilling these security requirements, fast and energy-efficient methods should be used. Although there are some recent works to make public key cryptography (PKC) practical to be used sensor nodes [4–6], symmetric cryptography and hash-based solutions are still more efficient to provide security in sensor networks [20, 21]. These solutions necessitate pairwise keys distributed among the sensor nodes prior to beginning of secure communication. The problem of distribution of keys to large number of sensor nodes is an active research area.

Key predistribution schemes [1, 7–12, 14] are shown to provide practical and efficient solutions. In such schemes, redundant amount of keys are stored in nodes' memory before deployment and a matching algorithm is processed between neighboring node pairs after the deployment. As a result of this match, some of the stored keys are used in secure communication of neighbors. If two neighboring nodes share a key, then a *secure link* exists between those nodes. Due to probabilistic nature of the scheme, some neighboring nodes may not share a key. In the literature, there are some location-aware approaches [8, 11, 15–17], where expected location information of sensor nodes is utilized, in order to improve the key sharing probability and the resiliency of the system by reducing the number of reused keys. In such location-aware approaches, it is assumed that nodes are prepared in small groups and deployed as bundles. Thus, the nodes in the same group have a large chance of being in the radio communication range of each other. Keys are stored in nodes such that nodes in the same or neighboring groups have common keys, but nodes in distant groups do not share any.

Blom's key management scheme [2] is used as a powerful tool in key predistribution schemes [9]. Blom's scheme shows a threshold property; until λ nodes are captured, the network is perfectly secure, but if $\lambda+1$ or more nodes are captured all secure links are compromised.

In this paper, we propose a zone-based and two-tier approach for key predistribution problem in sensor networks, where there are two types of sensor nodes with different capabilities: regular nodes and agent nodes. Agent nodes have larger memory and can share keys with agent nodes from neighboring zones. Agent nodes constitute a small part of sensor network. Regular nodes can establish secure links only with same-zone neighbors without intervention of agent nodes. We show that our approach significantly increases the resiliency of the system while still keeping the network connected via secure links to a large extent. Moreover the proposed scheme uses less flooding and consequently has less communication overhead as compared to rival schemes. Scalability is another remarkable feature of the proposed scheme; increasing the size of the sensor network field and the number of nodes together does not degrade the performance of the system.

The rest of this paper is organized as follows: in Section 2, we describe our key predistribution scheme. In Section 3, we provide a comparative analysis of our scheme. Finally, we provide some concluding remarks in Section 4.

2 Two-tier, location-aware key predistribution scheme

In our scheme, we exploit the deployment location knowledge of sensor nodes in order to improve the performance of key predistribution. If a group of sensor nodes is deployed at a deployment point, they will likely reside in close proximity with each other. We arrange target locations in a grid fashion and determine which bundle will be deployed at which target location. We name each cell of the grid as a *zone*. Before deployment, separate key spaces are created for each zone according to our key predistribution scheme. Using this method, we increase the average number of shared keys between nodes. The parameters and symbols used in this scheme are given in Table 1.

The key predistribution scheme consists of four phases; *predistribution phase*, *direct key establishment phase*, *hybrid key establishment phase*, *path key establishment phase*.

2.1 Zone based deployment model

We employ a classical zone based deployment model similar to the ones used in [8] and [22]. In zone based deployment models, the deployment area is divided into several contiguous zones. Moreover before the deployment, the nodes are grouped; and group-specific keys and/or keying materials are stored into the nodes of each group. After that, groups are deployed airborne (via a plane or helicopter) over the zones in one-group-per-zone basis. In this way, nodes that belong to a particular group are clustered in the proximity of a zone.

Table 1 Symbols and parameters

N	number of nodes in each zone
Z	number of zones and groups in the sensor network ($= Z_x \times Z_y$)
Z_x	number of rows in the sensor field
Z_y	number of column in the sensor field
ω	number of key spaces for each zone
τ	number of key spaces installed in a regular node
R	communication range of sensor nodes
A_z	number of agent nodes in each zone
s_{mn}	ID of n^{th} sensor node in zone m, $m=1 \,..\, Z$, $n=1 \,..\, N$
r_{mn}	resident point of node s_{mn}, $m=1 \,..\, Z$, $n=1 \,..\, N$
k_{mp}	ID of p^{th} key space in zone m, $m=1 \,..\, Z$, $p=1 \,..\, \omega$
Z_{ij}	ID of zone at i^{th} row, j^{th} column, $i=1 \,..\, Z_x$, $j=1 \,..\, Z_y$
d_{ij}	deployment point of zone Z_{ij}, $i=1 \,..\, Z_x$, $j=1 \,..\, Z_y$
G_{ij}	ID of group of nodes deployed at d_{ij}

In our deployment model, we divide the rectangular sensor field into a grid of $Z=Z_x \times Z_y$ equal sized zones. Before the deployment, sensor nodes are grouped into Z groups, each has N nodes. As will be explained in Section 2.2, the keys and keying materials are stored into the nodes also before the deployment. Center point of zone Z_{ij} is the deployment point, d_{ij}, of group G_{ij}, where $i=1 \,..\, Z_x$ and $j=1 \,..\, Z_y$. During the deployment, the nodes that belong to G_{ij} are dropped airborne over the deployment point d_{ij}. The actual location of a sensor node after deployment is its resident point, r_{mn}, where $m=1 \,..\, Z$ and $n=1 \,..\, N$. Resident points of sensor nodes in the same group follow the same probability distribution function. In our deployment model, we employ two-dimensional Gaussian distribution as in [8]. This distribution causes the sensor nodes, which are dropped at the same deployment point, tend to be closer to each other after the deployment. However, it is also possible to have some nodes that end up in neighboring zones after the deployment. This is the most difficult case in terms of key distribution since such a node may not find common keying material with some of the nodes in its deployed location. However, our key distribution scheme addresses this issue by the inter-zone path key distribution method explained in Section 2.5.

According to our deployment model described above, the nodes that bear the group ID x should have been deployed over zone with the same ID x. By combining these two facts and for the sake of simplicity of explanations, the characteristics of "being a member of group" and "being a member of a zone" will be used

synonymously from this point onwards. Among these two phrases, "being a member of a group" is the canonical one. Since a node may end up in a neighboring zone after the deployment or after a drift, "being a member of a zone" does not mean physically being in an area, but belonging to a particular group.

2.2 Predistribution phase and deployment

In key predistribution phase, we describe the method of how keys are distributed to nodes. This is performed before the nodes are deployed over the sensor field. We define two methods in the predistribution phase: *intra-zone key predistribution method* and *inter-zone key predistribution method*. In intra-zone key predistribution method, setup server distributes the keys required for establishing secure links between nodes from the same zone. This step applies for both regular nodes and agent nodes. In inter-zone key predistribution method, setup server distributes the keys to agent nodes for their secure communication with other agent nodes of neighboring eight zones.

In the *intra-zone key predistribution*, we adopted the method proposed in [9], which is for the whole sensor field, into a zone. The method in [9] and also our method are based on well-known Blom's key predistribution scheme [2], Using Blom's scheme, any two nodes having shares from the same matrix can compute a secret pairwise key. Setup server generates a single public matrix G, whose size is $(\lambda+1)\times N$. All $\lambda+1$ columns of $\mathbb{G}$ matrix are linearly independent. For each zone, setup server generates ω random and symmetric $\mathbb{D}$ matrices with size $(\lambda+1)\times(\lambda+1)$ and uses these matrices to compute ω $\mathbb{A}$ matrices. Size of each $\mathbb{A}$ matrix is $N\times(\lambda+1)$. Each $\mathbb{D}$ and $\mathbb{A}$ matrix pair make up a key space. Each key space has a unique ID, k_{mp}, where $1\le m\le Z$ and $1\le p\le\omega$. Sensors can use key space IDs to find out if they have common key spaces with their neighbors. Then, for each node s_{mn}, setup server picks τ key spaces and stores n^{th} row of $\mathbb{A}$ matrix and n^{th} column of $\mathbb{G}$ matrix to node s_{mn}.

In order for two neighboring nodes to compute a common key, they need to know each other's public columns in $\mathbb{G}$ matrix. As shown in [9], it is feasible to generate a public $\mathbb{G}$ matrix by using a single primitive element. Instead of storing $\mathbb{G}$ matrix columns, nodes only store a single primitive element. At the end of intra-zone key predistribution method, all nodes have τ rows with $\lambda+1$ elements and one primitive element stored in their memory.

In our method, each zone has distinct key spaces. This guarantees that the keys used in one zone are not used in another zone. In this way, the resiliency improves significantly as analyzed in Section 3.

As a unique feature of our method, in the *inter-zone key predistribution* method, we distribute random-pairwise keys to establish common keys between agent nodes. Before sensor deployment, setup server generates unique random pairwise keys for each agent node pair; there are only two copies of a pairwise key. For an agent node s_{mn}, setup server generates pairwise keys that s_{mn} shares with all agent nodes in neighboring zones of zone m. Then, these pairwise keys are stored in s_{mn} along with IDs of corresponding agent nodes.

Here one should notice that zone and consequently group IDs are encoded in node ID so that when the nodes exchange their IDs, they understand the zones/groups to which they belong.

Random pairwise keys have node-to-node authentication property and have perfect node capture resiliency, meaning that when a pairwise key is compromised by adversaries, only the secure link that compromised key is used, is affected.

Agent nodes will carry keys from both intra-zone and inter-zone key predistribution method. Thus, agent nodes must have larger memory as compared to regular nodes. Considering that there will be limited number of agent nodes in each zone, this is a practical approach.

After the keys and keying materials are predistributed, the nodes are deployed over zones as described in Section 2.1.

2.3 Direct key establishment phase

After deployment, sensor devices try to establish secure links with all of their neighbors. In direct key establishment phase, two neighboring nodes of the same group/zone compute shared keys with their neighbors. Here, we use a similar method as the one described in [9]. The two neighboring sensor nodes can be regular nodes or agent nodes. In order to find out if they share any key spaces, each node broadcasts a message containing the node's ID and the indices of the stored key spaces. If two neighboring nodes, s_{mn} and s_{mq}, are in the same group and share a common key space, then they can compute a pairwise key using Blom's scheme. s_{mn} can compute the pairwise key by using its private row from matrix $\mathbb{A}$ and s_{mq}'s column of public matrix $\mathbb{G}$, which s_{mn} can generate by using s_{mq}'s ID and the primitive root, which is already stored in every node. Similarly, s_{mq} calculates the same key using its private row and s_{mn}'s column of $\mathbb{G}$. This shared key is called the *direct key*.

Neighboring sensor nodes may belong to different groups/zones. If at least one of the nodes is a regular node, they cannot directly establish a secure link because they do not have any common key spaces. In Section 2.5, we describe an original method how two regular nodes from different zones can establish a secure link with the help of agent nodes. If both of the nodes are agent nodes from neighboring zones, they can easily establish a secure link by exchanging IDs. Each agent node can find the pairwise key shared with the other agent node just by using other node's ID.

After the direct key establishment phase, the entire sensor network forms a secure link graph in which two nodes can have an edge between them only if they are neighbors and they share a secret key.

In case of a drift of a sensor node, it may rerun this phase in order to establish keys with new neighbors of the same zone. If the drifted node is an agent node and if it faces with another agent node of a neighboring zone, the key distribution is trivial as explained above in this subsection. However, if a drifted agent node meets a regular node of the neighboring zone, it should run inter-zone path key establishment method as described in Section 2.5. Similarly, in case a regular node moves to a neighboring zone and faces with a node (regular or agent) that belongs to that zone, it should run inter-zone path key establishment method.

2.4 Hybrid key establishment method

Every regular node needs to have a contact with an agent node in order to perform inter-zone path key establishment that will be explained in Section 2.5. Direct key establishment phase can be used to establish direct keys between a regular node and an agent node of its zone. However, if a regular node has no agent node within its radio communication range (i.e. none of regular node's 1-hop neighbors is an agent node), they cannot run the direct key establishment procedures. In such as case, the nodes may run the hybrid key establishment method. In this method, the regular node tries to find an agent node within several hops range to establish a pairwise key.

Regular nodes may share key spaces with agent nodes even if they are several hops away from each other. If they can exchange their key space IDs over a secure path, they can compute their secret shared key as explained in Section 2.3. Hybrid key establishment method basically aims the exchange of such key space IDs over a secure path.

The hybrid key establishment method works as follows. Suppose a regular node, s_{mn}, where $1 \le m \le Z$ and $1 \le n \le N$, multicasts a query including its key space IDs to its secure neighbors with whom it shares a direct key. If s_{mn}'s secure neighbors have an agent node of its zone in their neighbor lists, they forward the query to the agent node. If there are no agent nodes in two hops, secure neighbors of s_{mn} forward the query to their secure neighbors and this flooding[1] of queries goes on until either a hop-limit is reached or an agent node of that zone is found. If the secure link graph is connected, s_{mn} eventually finds an agent node. If more than one agent node is found in this way, then the closest one is preferred. In this method, not only a key is exchanged, but also a secure path is established between s_{mn} and its closest agent node. This secure path is later utilized in inter-zone path key establishment phase.

An example of hybrid key establishment method is shown in Fig. 1. Here the regular node s_{ib} has an agent node s_{ia} which is 3-hops away from it.

It is possible that regular node s_{mn} does not share any key spaces with any of the agent nodes in its zone. In this case, a path key can be established between s_{mn} and its nearest agent node using the method explained in Section 2.5.

In the case of a regular node is drifted within the network, hybrid key establishment still works provided that there is a secure path between this regular node and its agent node. Similarly, when an agent node is drifted, its regular nodes can still use hybrid key establishment if there is a secure path towards this agent node. As will be discussed in Section 3.3, our scheme shows a good global connectivity performance, meaning that only less than 1% of all nodes are disconnected from other nodes. If a node shares key with at least one of its neighbors after a drift, most probably this node will find a path to other nodes. This is valid for both regular and agent nodes. That is why the problem of existence of a secure path between a regular node and an agent node is reduced to keep the drifted nodes cryptographically connected. This connectivity can be achieved via direct and path key establishment mechanisms explained in Sections 2.3 and 2.5, respectively. Among these, inter-zone path key establishment mechanism is particularly important for a drifted node if it moves to a neighboring zone.

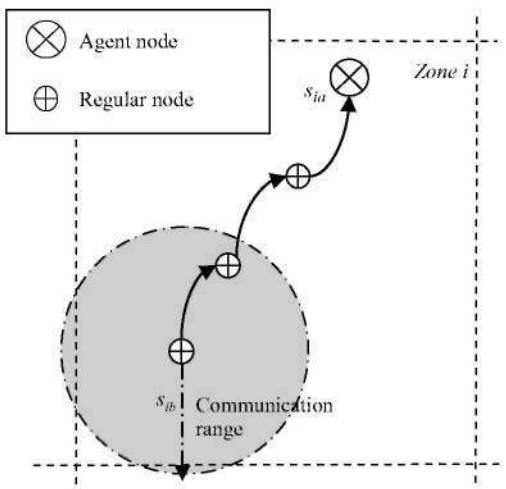

Fig. 1 Regular node s_{ib} establishes a pairwise key with agent node s_{ia} using *hybrid key establishment method*

[1] In this paper, we use *restricted* flooding in which a query is broadcasted by a node only once: when the same query is received again by the same node, it is simply dropped and not broadcasted. In this way, the communication cost caused by flooding is reduced as compared to unrestricted flooding in which a particular node unnecessarily broadcasts duplicate queries.

2.5 Intra-zone and Inter-zone path key establishment phases

After direct key establishment phase, a sensor node, s_{mn} may end up in a case where it cannot find any shared key spaces with one or more of its neighbors. In this case, s_{mn} tries to find secure paths to such neighbors with the help of its secure neighbors. The process of establishing a secure link over a secure path between same zone nodes is called *intra-zone path key establishment.*

The process works as follows. Assume node s_{mn} of zone Z_m does not have a secure link with its neighbor node s_{mp}. Node s_{mn} floods a query to other nodes to see if they have secure links with node s_{mp}. If at some hop level any of the neighbors, say s_{mq}, has such a secure link, then s_{mq} generates a random key and sends this key to both node s_{mn} and s_{mp} over secure links. Then, s_{mq} removes this random key from its memory.

When node s_{mn}'s neighbor, s_{tk}, is from a neighboring zone, s_{mn} needs an agent node to communicate securely with s_{tk}. That is why every regular node needs a secure path to its nearest agent node before initiating *inter-zone path key establishment process*. Assuming both s_{mn} and s_{tk} have direct or hybrid links with an agent node, inter-zone path key establishment process works as follows:

1. They exchange their and their nearest agent node's ID.
2. One of the regular nodes, say s_{tk}, sends IDs received from the other node to its nearest agent node over a secure link.
3. Since s_{mn} and s_{tk} are from neighboring zones, their agent nodes must share a pairwise key, as explained in Section 2.2. Agent nodes can easily find out their shared pairwise key, K_p, via a simple lookup. Node s_{tk} has either a direct or hybrid key, K_s, to its agent node. Node s_{tk}'s agent node generates a random key, K_r, and encrypts it with K_p as $E_{Kp}\{K_r\}$. Then s_{tk}'s agent node prepares and sends the message $E_{Ks}\{K_r, E_{Kp}\{K_r\}\}$ to s_{tk} over a secure path or secure link.
4. Node s_{tk} decrypts the message and retrieves K_r. Then it sends $E_{Kp}\{K_r\}$ to its neighbor, s_{mn}.
5. Node s_{mn} sends the message, $E_{Kp}\{K_r\}$, to its agent node. The agent node decrypts $E_{Kp}\{K_r\}$ and sends K_r back to s_{mn} over a secure link or secure path.
6. Now both s_{mn} and s_{tk} shares the same key K_r.

As an example, we provide the inter-zone path key establishment process between two nodes, s_{ib} and s_{jb}. Nodes s_{ib} and s_{jb} belong to neighboring zones and they've already established hybrid keys with their nearest agent nodes, s_{ia} and s_{ja}, respectively. The example is shown in Fig. 2.

1. Sensor node s_{jb} initiates the inter-zone path key establishment process by sending message $M_1=\{s_{jb}, s_{ja}\}$ to node s_{ib}
2. Sensor node s_{ib} receives M_1 and sends back $M_2=\{ s_{ib}, s_{ia} \}$ to s_{jb}
3. Assume that agent node s_{ia} and regular node s_{ib} has already established a secure link and share a direct key, K_d. Sensor node s_{ib} sends $M_3=E_{Kd}\{\{ s_{jb}, s_{ja} \}\}$ to its agent node s_{ia}.
4. Agent node s_{ia} looks up pairwise key, K_p, that it shares with s_{ja}. Then s_{ia} generates a random key, K_r, to be used as s_{ib} and s_{jb}'s shared secret key. After that, agent node s_{ia} sends encrypted message $M_4=E_{Kd}\{K_r, E_{Kp}\{K_r\}\}$ to s_{ib}.
5. Sensor node s_{ib} receives and decrypts M_4 and retrieves K_r and $E_{Kp}\{K_r\}$. Then, s_{ib} sends $M_5=E_{Kp}\{K_r\}$ to s_{jb}.
6. Sensor node s_{jb} receives M_5 but cannot decrypt it because s_{ib} and s_{jb} do not yet share any key. In addition, s_{jb} does not have a direct link with agent node s_{ja}; s_{jb} can reach agent node s_{ja} in two hops with the help of s_{jc}. Assume that s_{jb} and s_{jc} has already

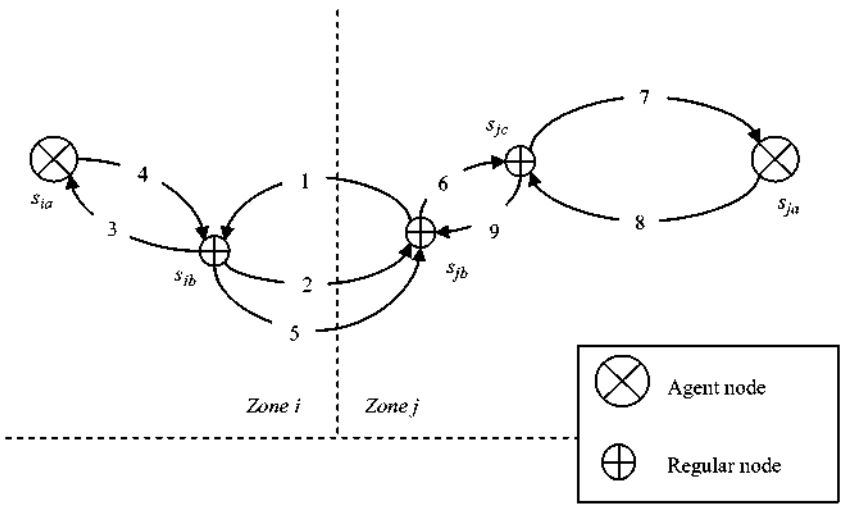

Fig. 2 Example case: Two neighboring regular nodes, s_{ib} and s_{jb}, from different zones establish a secure link through inter-zone path key establishment process

established secure link using direct key establishment method and share a direct key, K_{d1}. Sensor node s_{jb} sends $M_6=E_{Kd1}\{E_{Kp}\{K_r\}\}$ to s_{jc}.

7. Sensor node s_{jc} receives and decrypts M_6, then retrieves $E_{Kp}\{K_r\}$. After that, s_{jc} encrypts $E_{Kp}\{K_r\}$ with K_{d2}, shared direct key with s_{jc} and agent node s_{ja}. s_{jc} sends $M_7=E_{Kd2}\{E_{Kp}\{K_r\}\}$ to s_{ja}.
8. Agent node s_{ja} receives and decrypts M_7, retrieves $E_{Kp}\{K_r\}$. Then using the shared pairwise key between two agent nodes, s_{ja} and s_{ia}, s_{ja} retrieves K_r. Assume that agent node s_{ja} and s_{jb} shares a secret key, K_{d3}, generated using the hybrid key establishment method, explained in Section 2.4. Then, s_{ja} encrypts K_r with E_{Kd3} and gets $E_{Kd3}\{K_r\}$. Agent node s_{ja} re-encrypts the message with E_{Kd2} and sends $M_8=E_{Kd2}\{E_{Kd3}\{K_r\}\}$ to sensor node s_{jc}.
9. Sensor node s_{jc} receives and decrypts M_8, then retrieves $E_{Kd3}\{K_r\}$. After that, s_{jc} encrypts $E_{Kd3}\{K_r\}$ with K_{d1}, shared direct key between s_{jc} and s_{jb}. Regular node s_{jc} sends $M_9=E_{Kd1}\{E_{Kd3}\{K_r\}\}$ to s_{jb}.
10. Lastly, sensor node s_{jb} receives and decrypts M_9, then retrieves $E_{Kd3}\{K_r\}$. By decrypting $E_{Kd3}\{K_r\}$, s_{jb} retrieves K_r. Using K_r, s_{ib} and s_{jb} can securely communicate with each other.

Both intra-zone and inter-zone path key establishment methods can be rerun in case a node is drifted within the network due to external conditions. The uses of these mechanisms for a drifted node are not so different than the normal use. After a drift, if a node meets another node of the same zone, they may use intra-zone path key establishment. If the drifted node faces with a node of a neighboring zone, then inter-zone path key establishment is used. The only difference in the drifting case is that the drifted node should not get isolated. In other words, the drifted node should remain connected in such a way that it shares keys with some neighbors. The reason of this requirement is that both intra and inter path key establishment mechanisms need to have at least one secure link. In order to keep the drifted node connected, it should continuously check for the new neighbors and whenever it sees a new neighbor, it should establish a key with it. In this way, even if some secure links are dropped due to mobility, using the already established secure links, new secure links are created and the connectivity of the mobile node with the rest of the network is sustained.

The abovementioned discussion of mobile nodes is valid for both regular and agent nodes. In case of an agent node moves towards a neighboring zone, it can still be connected via inter-zone path key establishment method. Actually, it is easier for an agent node to run inter-zone path key establishment as compared to a regular node. The reason is that a regular node needs to have shared key with its agent node as a prerequisite of inter-zone path key establishment; however, for an agent node there is no such requirement.

As discussed above, our scheme addresses the mobility of nodes into the neighboring zones only. If the nodes are drifted further to other distant zones, our scheme works if and only if agent nodes of these distant zones share keys. Currently, our scheme does not provide such a sharing. However, by increasing the key storage memory, pairwise keys for agent nodes of distant zones could be predistributed prior to deployment. Such a high mobility case is not addressed in this paper and is left as a future work.

3 Performance evaluations

In order to evaluate the performance of our scheme, various simulations are performed in Matlab®. We used the well-known metrics such as local connectivity, global connectivity, communication cost, and resilience against node compromise. We also simulated some of the well-known key predistribution schemes [1], [8], and [9] for comparison purposes.

3.1 System parameters

In our analysis and simulation, we use the following configuration.

- Deployment area is 1,000 m × 1,000 m
- Deployment area is divided into 10×10 zones, i.e. $Z_x=Z_y=10$ and $Z=100$
- Total number of sensor nodes is 10,000 and there are 100 nodes in each zone, i.e. $N=100$.
- Communication range, R, for each node is 40 m. The nodes are assumed to be static.

3.2 Local connectivity

Local connectivity can be referred as the probability of two neighboring nodes sharing at least one key space, in other words having a direct secure link. Assuming that key spaces are homogenously distributed among sensor nodes, local connectivity can also be defined as the average number of secure neighbors of a node. This probability is denoted as P_{local}. In Fig. 3, local connectivity values of our scheme and Du et al.'s scheme [9] are shown. It can be observed that the ratio τ/ω is the determiner P_{local}. As τ increases and ω decreases, the probability that two neighboring nodes share at least one key space increases. In this analysis, the ω values of Du et al.'s scheme is taken 100 times larger than our scheme in order to equalize the total number of key spaces in the whole sensor network.

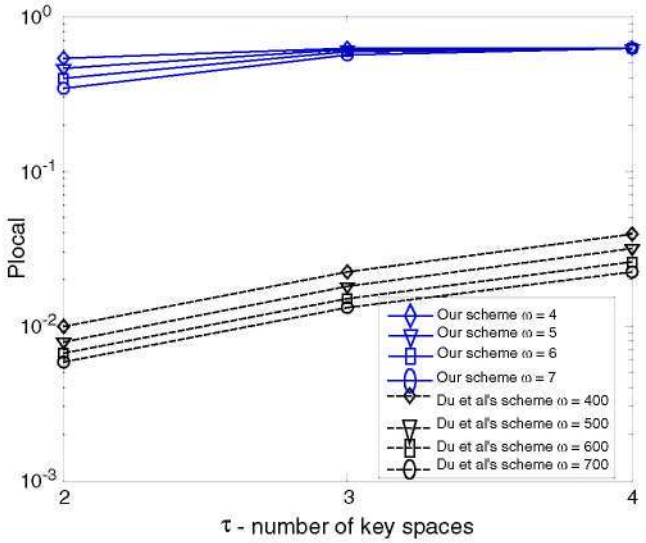

Fig. 3 Local connectivity, P_{local}, vs. τ, number of key spaces installed in a node

Figure 4 shows local connectivity values obtained from simulation results of our scheme and Du et al.'s scheme using deployment knowledge [8] (from now on we call this scheme "Du et al.'s scheme 2"). Their approach is a modified version of Eschenauer and Gligor's scheme [1]. They improve the scheme in [1] by using deployment knowledge on a grid environment.

In Fig. 4, we simulated our scheme for various values of $\lambda+1$ and τ values. We took 15, 25 and 35 as $\lambda+1$, and 2, 3 and 4 as τ. For our scheme, $\tau\times(\lambda+1)$ gives the number of keys in a node which is shown on the horizontal axis of Fig. 4. When $\lambda+1$ is 15 and τ is less than 4, our scheme has better local connectivity than Du et al.'s scheme 2. However, P_{local} of our scheme does not increase more than 0.6209. However, P_{local} of Du et al.'s scheme 2 reaches 0.9522 when number of keys is as high as 150. Local connectivity for our scheme stops increasing after a specific value, because regular nodes cannot establish direct secure links with their different-zone neighbors, whereas in Du et al.'s scheme 2, nodes have the capability to share keys with nodes from neighboring zones. Although it seems here that our scheme has a drawback here for large number of keys, since it is possible to reach good global connectivity and resiliency figures with 50–60 keys, as discussed in subsequent sections, larger amount of keys only marginally affects the overall performance of the system at a high cost of larger memory at tiny sensor nodes.

3.3 Global connectivity

Even if a sensor node cannot establish a direct secure link with its neighbor, it is possible to establish a link via path key establishment phases provided that the node has a secure path to this neighbor. If we generalize this to all sensor nodes, in order to establish secure links via path key establishment phases, the network must be *securely* connected after the direct key establishment phase. Global connectivity is the measure of this *secure* connectedness. Global connectivity is computed by finding the ratio of the largest securely connected block of nodes (obtained after direct key establishment phase) over total number of nodes. Global connectivity also indicates the amount of wasted nodes. If some nodes have no secure connection with the main block of sensor nodes, then they cannot contribute to the sensor network securely. For example, consider 0.99 global connectivity for a sensor network. This means 99% of all nodes can establish direct or path keys among themselves; however, 1% of the nodes cannot reach the rest of the network in a secure way.

Figure 5 shows global connectivity of our scheme for $\tau=2, 3, 4$ and $\omega=4, 5, 6$. Simulation results indicate that even in the worst case where $\tau=2$ and $\omega=6$, global connectivity is higher than 0.99, which means more than 99% of nodes securely join and contribute to the sensor network.

3.4 Communication cost

In this section, communication overhead of our key predistribution scheme, when two neighboring nodes cannot establish a direct secure link, is examined. In our scheme, a sensor network incurs most of communication cost during three operations: intra-zone path key establishment, hybrid key establishment and inter-zone path key establishment. During intra-zone path key and hybrid key establishment processes, flooding is used in broadcast and multicast manner, respectively.

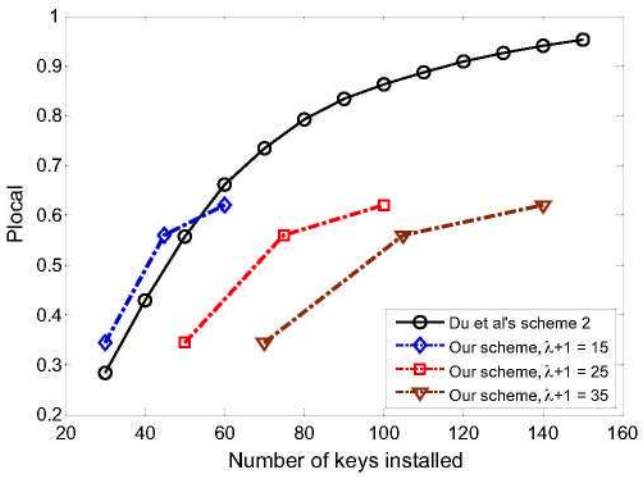

Fig. 4 Local connectivity for Du et al.'s scheme 2 [8] and our scheme. For our scheme $\omega=7$, $\tau=2, 3, 4$, and $\lambda+1=15, 25, 35$. Number of keys in a sensor node is calculated as $\tau\times(\lambda+1)$

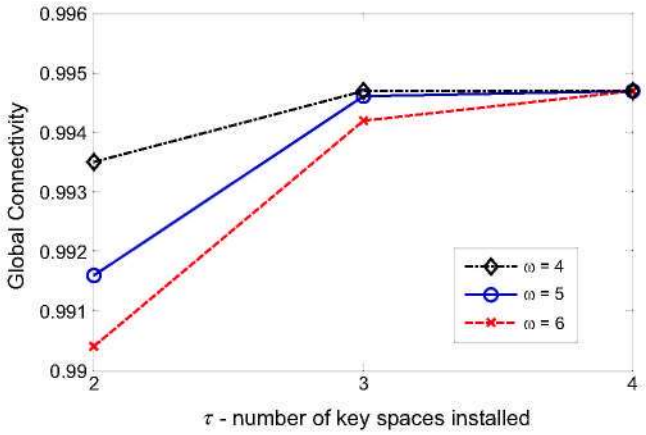

Fig. 5 Global connectivity of our scheme

We first determine average number of hops required to connect two neighboring nodes using intra-zone path key establishment. Figure 6 illustrates number of hops and connectivity values of corresponding secure link graphs for various τ and ω combinations. It can be observed from Fig. 6 that when τ/ω ratio is high, a node can establish direct links with most of its same-zone neighbors. For example, when τ is 3 and ω is 6, a node can reach 0.9503 of its same-zone neighbors in one hop, and the rest in two hops. Although we could not show here for space limitations, the performances of flooding based path key establishment of our scheme and Du et al.'s scheme 2 [8] are similar.

The number of hops, that a regular node can reach its nearest agent node, is an important indicator of network connectivity and an important parameter in overall communication cost. We show in Fig. 7 that majority of regular nodes can reach their nearest agent nodes in only one hop when the number of node agents in a zone, A_z=10.

Here it should be noted that while a hybrid key is being established, flooding is required only for one time. Then the same path can be used for all subsequent inter-zone path key establishment processes. This is an important advantage of our scheme.

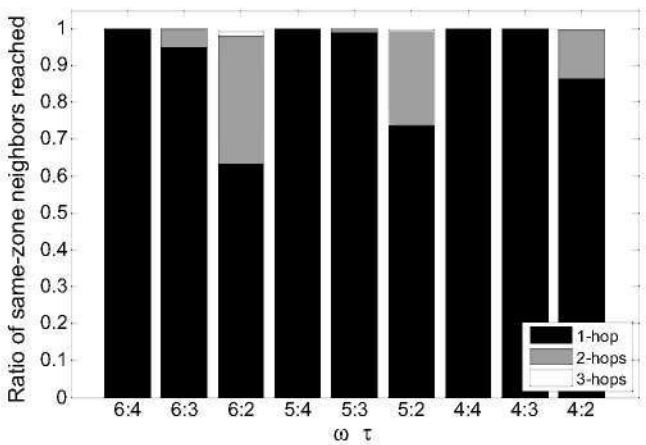

Fig. 6 Communication overhead for intra-zone path key establishment in our scheme

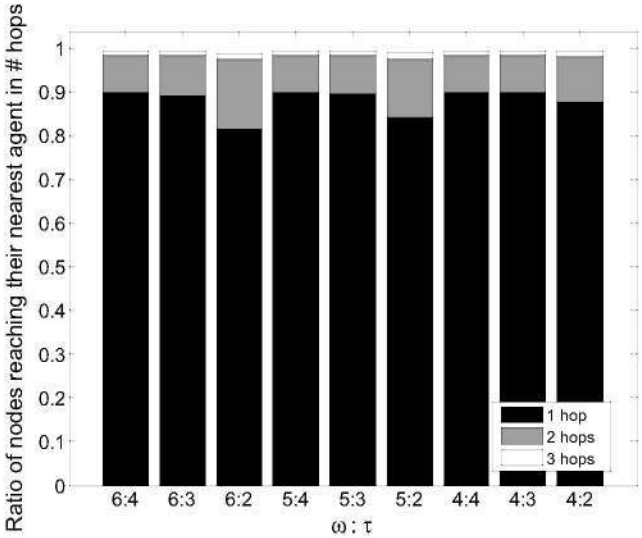

Fig. 7 Ratio nodes reaching their nearest zone agent in i hops when A_z=10 (for our scheme)

When two neighboring regular nodes are from different zones, they try to establish a secure link by inter-zone path key establishment process, as discussed in Section 2.5. Assuming that one of the regular nodes is h_1 hops away from its zone agent and hop length between the other node and its zone agent is h_2, total number of messages exchanged during inter-zone path key establishment process can be found as:

$$2(h_1 + h_2) + 3 \tag{1}$$

We calculate the number of messages exchanged for each inter-zone path key. Figure 8 illustrates the ratio of inter-zone path keys established by exchanging different amounts of protocol messages. For example, when ω=6, τ=2 and A_z=5, 80 % of all inter-zone path keys are established by exchanging 9 or less protocol messages. Maximum number of messages required in order to establish all inter-zone path keys is 13 when ω=6, τ=2 and A_z=10. Here one may argue that the number messages is quite larger than the intra-zone path key establishment process. However, it should be noted that inter-zone path key establishment does not make flooding which may exponentially increase the number of messages distributed in the network. The use of flooding in our scheme and Du et al.'s scheme 2 [8] will be examined in more detail in Section 3.6.

3.5 Resiliency against node capture

The most obvious attack against a sensor network is capturing sensor nodes. We will assume when a node is captured, all of its cryptographic material is compromised.

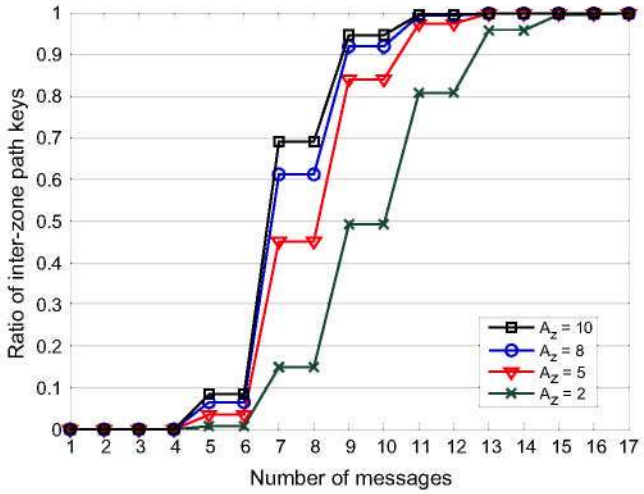

Fig. 8 Communication cost for inter-zone path key establishment in our scheme, where ω=6 and τ =2

Using those compromised material, attacker can also compromise some additional links that use the same material. A key distribution scheme's resiliency against node capture can be defined as the ratio of additional compromised links over total number of links except those of captured nodes. The smaller this ratio is the more resilient network. One possible way to protect keys inside a sensor node is to tamper-proof the device. However tamper-proofing is both costly [18] and is not perfectly safe [13].

For a key space to be compromised, $\lambda - 1$ nodes carrying shares from that key space must be compromised [2]. An attacker with λ shares from the same matrix cannot gain any extra information about that key space and cannot learn private shares of nodes that are not captured.

In Fig. 9, we show node capture resiliency of our scheme, Du et al.'s scheme 2 [8] and Du et al.'s scheme [9]. For our scheme, ω=7, τ=3, λ=1=17 and P_{local} =0.5605. For Du et al.'s scheme 2, m=50, S_c=1000 and P_{local}= 0.5569. For Du et al.'s scheme, ω=43, τ=4, λ=1=13 and P_{local} =0.56. As shown from these figures, three of the systems are compared using similar values for the number of keys per node and local connectivity.

It can be observed from Fig. 9 that both Du et al.'s scheme 2 and our scheme have substantially better resiliency than Du et al.'s scheme [9]. The most important reason for such a difference is that scheme in [9] does not utilize deployment knowledge.

Our scheme has stronger resiliency than Du et al.'s scheme 2, especially against small-scale attacks. When number of captured nodes is less than 2000, our scheme causes zero or negligible number of additionally compromised links. However, in Du et al.'s scheme 2, an adversary can compromise 62% of secure links by capturing only 2,000 nodes. The reason is that our scheme is based on Blom's scheme and in Blom's scheme an attacker can gain no information on a key space with less than λ=1 shares. Therefore, attacker must capture a substantial number of nodes before compromising any additional links. However, with Du et al.'s scheme 2, when an attacker captures only one node, he can start to compromise additional secure links. Another reason that makes our scheme more resilient than Du et al.'s scheme 2 is the independence of the key spaces in different zones. In this way, when a key space is compromised, only the current zone is affected; the nodes in any other zone are not.

3.6 Use of flooding and its effect in communication cost

Restricted flooding is an inevitable tool in several key establishment mechanisms proposed in the literature. When flooding method is used during establishment of a pairwise key, a node broadcasts a query and each neighboring node forwards this query to all their next hop neighbors. Thus during flooding, excessive amount of messages are transmitted and received. Apparently, flooding incurs heavy communication cost on the sensor network. Eschenauer and Gligor's scheme [1], both Du et al.'s schemes [8, 9] use restricted flooding of key establishment request messages during path key establishment phases. Similarly, our scheme also uses restricted flooding during intra-zone path key and hybrid key establishment phases. However, as discussed in this subsection, the use of flooding is more restricted in our scheme.

Figure 10 depicts the amounts of different types of keys established in our scheme which are results from simulations performed using parameters described in Section 3.1. In our scheme, only intra-zone path keys and hybrid keys are established via flooding. In Fig. 10, sum of intra-zone path keys and hybrid keys are shown as "keys using flooding". Communication cost of establishing other keys (i.e. direct keys and inter-zone path keys) are less than establishing keys using flooding. As can be seen from

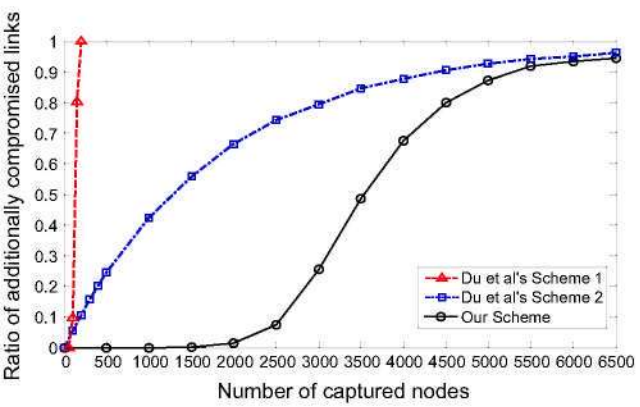

Fig. 9 Ratio of additionally compromised links vs. number of nodes captured for our scheme, Du et al.'s scheme 2 [8] and Du et al.'s scheme [9]. P_{local} is approximately 0.56 for al schemes

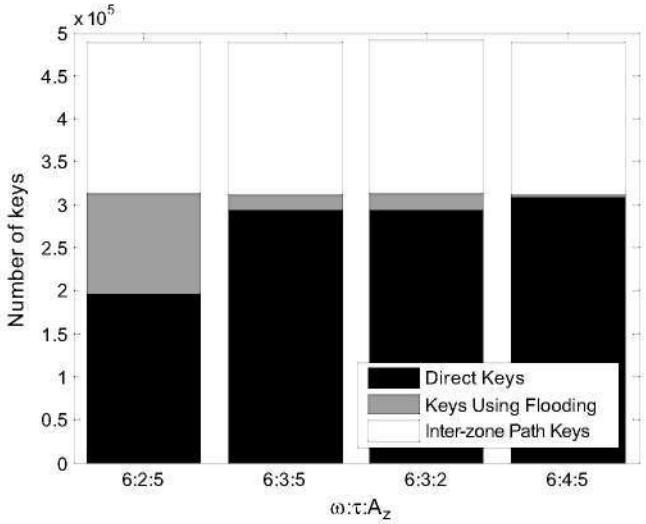

Fig. 10 Number of keys established in our scheme. "Keys Using Flooding" shows total number of intra-zone path keys and hybrid keys

Fig. 10, keys established via flooding cover only a small ratio of all keys. As ω/τ ratio increases, less flooding is used. Moreover, increasing A_z, decreases the number of keys established via flooding.

In Du et al.'s scheme 2 [8], path keys are established via flooding. In Fig. 11, we present simulation results for the number of different types of keys established in a sensor network with 10,000 nodes using Du et al.'s scheme 2. As m, number of keys stored in a node, increases, number of direct keys increases and, thus, less flooding is required. It can be observed from Fig. 11 that, when m is less than 50, majority of the pairwise keys are path keys. Thus, majority of the keys are established using flooding.

In Fig. 12, we compare use of flooding in our scheme versus Du et al.'s scheme 2 [8]. In our scheme, number of keys, m, is calculated as $m = (\lambda + 1)\tau$. In Fig. 12, we take $\lambda+1$ as 17, which is the value used in node capture resiliency simulations, and plot the change of number of keys using flooding with respect to τ values 2, 3 and 4 (that corresponds to m values 34, 51, and 68, respectively). As it can be seen from Fig. 12, for all m values, our scheme uses less flooding than Du et al.'s scheme 2. Figures 10 and 11 show that total numbers of keys in both schemes are approximately equal. Thus, the ratio of keys established via flooding in our scheme is less than that of Du et al.'s scheme 2. The rest of the keys are either direct keys or inter-zone path keys in our scheme. In Du et al.'s scheme 2, the rest of the keys are direct keys.

Establishment of inter-zone path keys causes more messages to be transferred in the network as compared to direct keys. However, the amount of messages transferred is still less than the total messages transferred in case of a flooding-based key establishment phase. In our simulation setting where there are 100 nodes with 40 m of communication range in a 100 m×100 m zone, the average number of neighbors is calculated as approximately 50. This means, each broadcast is received by 50 nodes in one hop. This value exponentially increases in multi-hop. However, in our inter-zone path key establishment, the communication is point-to-point (i.e. no flooding is used) and finishes in at most 17 messages. This conclusion and the fact that the amount of keys that need flooding in our scheme is less than Du et al.'s scheme 2 imply that overall communication cost of our scheme is cheaper than Du et al.'s scheme 2.

3.7 Scalability analysis

In a scalable key predistribution scheme, size of the sensor network should be increased without incurring any significant additional overhead on the sensor nodes and on the

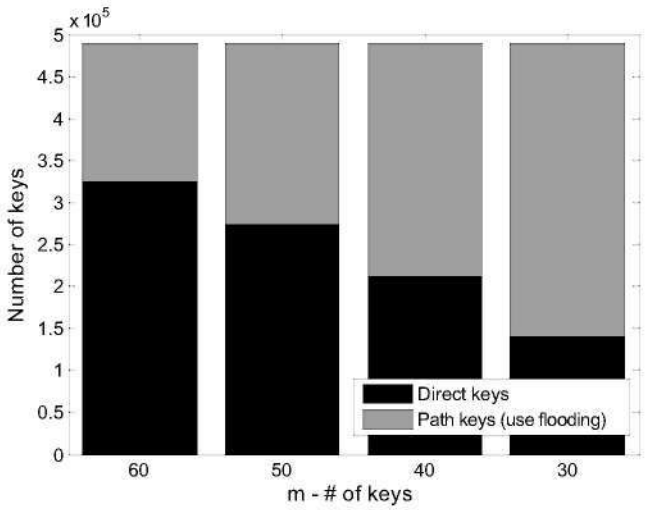

Fig. 11 Number of keys established in Du et al.'s scheme 2. Path keys use flooding

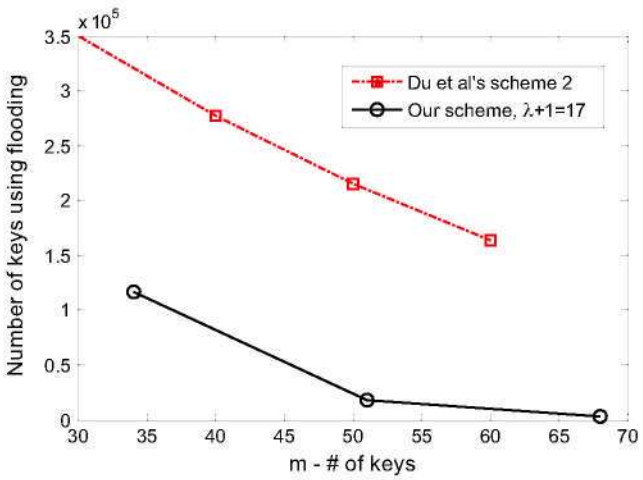

Fig. 12 Number of keys established using flooding in our scheme and Du et al.'s scheme 2

system. In this subsection, we increase, Z, the number of zones in simulated sensor network from 100 to 1,000 while keeping the number of nodes per zone and the zone size fixed. In this way, we increase the sensor field size and the number of nodes in the system. Our analyses, as detailed below, show that our scheme does not incur additional overhead in all metrics when the field size and number of nodes increase in this way.

We perform simulations of our scheme with the deployment region divided into 40×25 grid (i.e. $Z=1000$). In this way we increase the sensor field size tenfold as compared to other simulations given in previous subsections. Moreover, the total number of nodes is increased at the same rate (i.e. tenfold). The rest of the parameters are the same as the parameters given in Section 3.1. Local connectivity of our scheme when Z, number of zones, is equal to 100 and 1,000 is plotted in Fig. 13. Similarly global connectivity performance is depicted in Fig. 14. Results show that performance of our scheme does not change much when the sensor network field and the number of nodes grow. When the number of zones and nodes are increased tenfold, the ratio of secure neighbors a node can reach stays almost the same. Thus, adding new zones and nodes to the sensor network does not increase the memory cost on sensor nodes in our scheme.

In Fig. 15, we show the communication cost of intra-zone path key establishment for $\omega:\tau$ equal to 6:4, 6:3 and 6:2. We analyze the communication cost of intra-zone path key establishment by computing the ratio of same-zone neighbors a node can reach after i hops, where $i=1, 2, 3$. The first three bars show the communication cost when $Z=1000$ and the last three show the communication cost when $Z=100$. It can be observed that enlarging the sensor network by adding new zones and nodes do not increase the communication cost of intra-zone path key establishment.

In Fig. 16, we show the resiliency of our scheme against node capture for different sensor network sizes. It can be

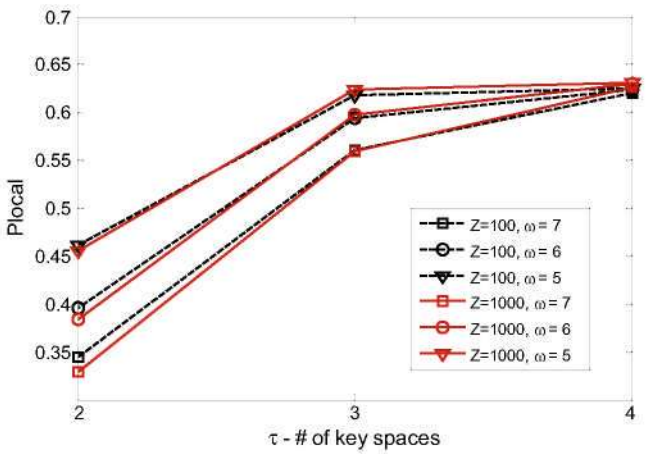

Fig. 13 Local connectivity for our scheme when $Z=100$ and $Z=1000$

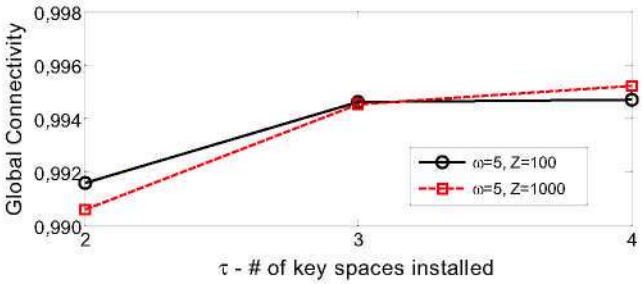

Fig. 14 Global connectivity of our scheme when $Z=100$ and $Z=1000$

seen from Fig. 16 that the ratio of compromised links are almost the same for $Z=100$ and $Z=1000$ when the ratio of captured nodes are the same. Thus, increasing the size of the sensor network does not significantly affect the node capture resiliency of our scheme.

After these observations, we can say that in all performance metrics that we analyzed, our scheme is scalable such that increasing the size of the sensor network field by adding new zones and new nodes in those zones does not affect the performance of the system and does not bring extra overhead on the sensor nodes.

3.8 Effect of communication range

We also analyze the effect of communication range of sensor nodes on local connectivity and global connectivity via simulations. In the analyses mentioned in previous sections, the communication range, R, was taken as 40 m. In this section, we compare results obtained using three different communication range values, $R=30$, $R=40$ and $R=50$ m.

In this section, as in Fig. 4, we analyze local connectivity for various values of $\lambda+1$ and τ. The results are depicted in Fig. 17. We take 15, 25 and 35 as $\lambda+1$; 2, 3 and 4 as τ. The value of ω is taken as 7. Number of keys in a node, which is

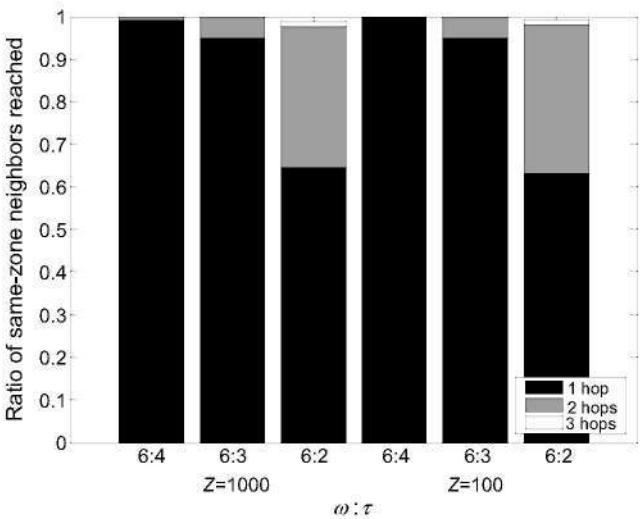

Fig. 15 Ratio of same-zone neighbors reached when $Z=1000$ and $Z=100$

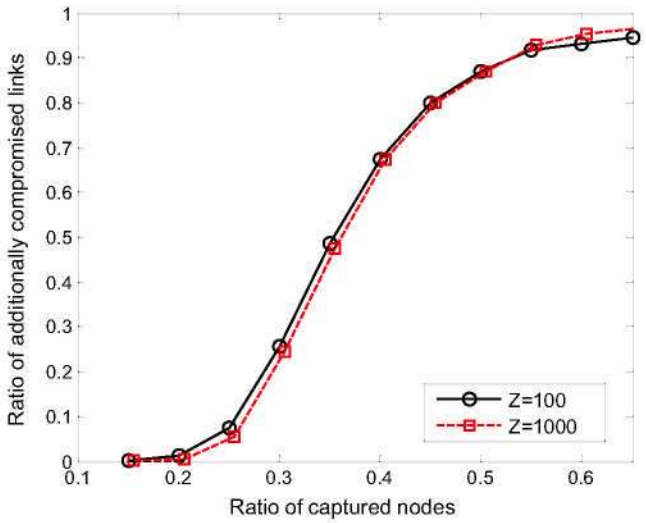

Fig. 16 Ratio of additionally compromised links for our scheme when $Z=100$ and $Z=1000$

calculated as $\tau\times(\lambda-1)$, is shown on the horizontal axis. As mentioned above, local connectivity values are depicted for three different communication range values, 30 m, 40 m and 50 m. As shown in Fig. 17, as the communication range decreases, paradoxically, local connectivity increases. The reason of this behavior is that decreased communication range automatically reduces the number of distant neighboring nodes that belong to different zones. Consequently, a particular node communicates with less number of nodes with which it does not share a key due to being part of different zones. In this way, the total number of neighbors of a node decreases, but the remaining nodes are mostly from the same zone. This increases the local connectivity.

Because of similar arguments, local connectivity decreases as the communication range increases. This is due to the fact that the number of neighbors, which belong to different zones, increases with increased communication range. Clearly, the number of cryptographically connected neighbors does not increase significantly in this setting. Therefore, local connectivity decreases.

One may argue that since reduced communication range increases local connectivity, it is more favorable to set the sensor nodes with small communication range parameters. However, this is not correct. Reduced communication range increases local connectivity to some extent, but it also reduces the number of secure neighbors according to our analyses. This, in turn, would reduce global connectivity of the network since some nodes become disconnected. In order to justify this thesis, we perform simulations. In these simulations, we analyze the global connectivity of the network under different communication range values, 30 m, 40 m and 50 m. We take $\omega=6$ and $\tau=2, 3, 4$. The results are depicted in Fig. 18. As shown in this figure, as the communication range decreases, global connectivity also decreases. This proves our thesis mentioned above.

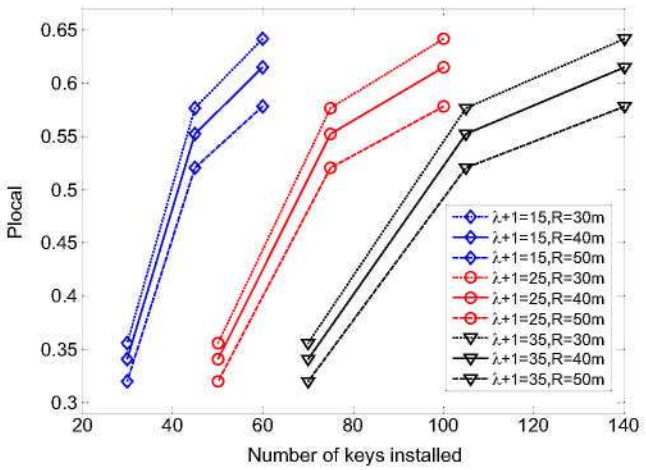

Fig. 17 Local connectivity for different communication ranges. $R=30$ m, 40 m, 50 m. $\omega=7$, $\tau=2, 3, 4$, $\lambda+1=15, 25, 35$. Number of keys in a sensor node is calculated as $\tau\times(\lambda+1)$

4 Conclusions

In this paper, we presented a two-tier random key predistribution scheme for sensor networks. In our scheme, we used a zone-based approach, in which each zone has its own distinct key spaces. Secure links between zones are established through agent nodes, which are higher capacity nodes. We utilized Blom's scheme [2] for key establishment among the nodes of the same zones.

Our scheme achieves high local and global connectivity values while consuming minimal memory. The communication cost of our scheme is within practical limits. Our scheme uses limited flooding during path key establishment. We showed that by using a two-tier approach, our scheme achieves substantially strong node capture resiliency. Ratio of additionally compromised links when 2,000 nodes (out of 10,000 total nodes) are captured is almost

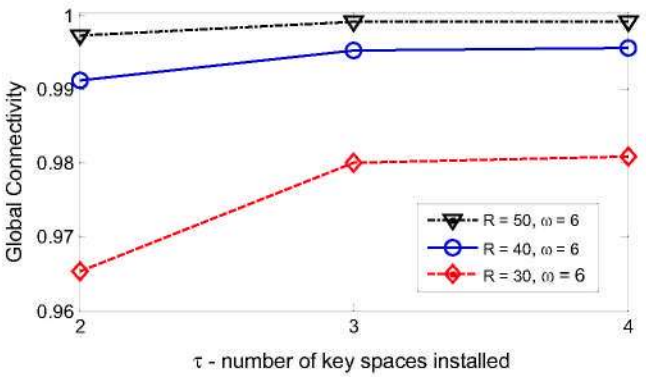

Fig. 18 Global connectivity of our scheme under different communication range values

zero. Our scheme is proven to be scalable such that increasing the size of the sensor deployment field and the number of nodes, the performance metrics do not worsen.

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

References

1. Eschenauer L, Gligor VD (2002) A key-management scheme for distributed sensor networks. Proceedings of the 9th ACM Conference on Computer and Communications Security, November 2002, pp 41–47
2. Blom R (1985) An optimal class of symmetric key generation system. Advances in Cryptology—Eurocrypt'84. LNCS, 209, 335–338. Springer
3. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. IEEE Commun Mag 40(8):102–114. doi:10.1109/MCOM.2002.1024422
4. Malan D (2004) Crypto for tiny objects. Harvard University Technical Report TR-04-04
5. Gaubatz G, Kaps J, Sunar B (2004) Public keys cryptography in sensor networks—revisited. Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS), LNCS 3313:2–18, Springer
6. Watro R, Kong D, Cuti S, Gardiner C, Lynn C, Kruus P (2004) Securing sensor networks with public key technology. Proceedings of the 2nd ACM Workshop on Security of Ad hoc and Sensor Networks, October 2004
7. Chan H, Perrig A, Song D (2003) Random key predistribution schemes for sensor networks. IEEE Symposium on Research in Security and Privacy 197–213
8. Du W, Deng J, Han YS, Chen S, Varshney P (2004) A key management scheme for wireless sensor networks using deployment knowledge. Proceedings of IEEE INFOCOM'04, March 2004
9. Du W, Deng J, Han YS, Varshney P (2003) A pairwise key predistribution scheme for wireless sensor networks. Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03), pp 42–51
10. Liu D, Ning P (2003) Establishing pairwise keys in distributed sensor networks. Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03), pp. 52–61, October 2003
11. Liu D, Ning P (2003) Location-based pairwise key establishments for static sensor networks. 2003 ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN '03), pp. 72–82
12. Zhu S, Setia S, Jajodia S (2003) LEAP: Efficient security mechanisms for large-scale distributed sensor networks. Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03), pp. 62–72.
13. Anderson R, Kuhn M (1996) Tamper resistance—a cautionary note. Proceedings of the Second Usenix Workshop on Electronic Commerce, pp. 1–11
14. Lin HY, Pan DJ, Zhao XX, Qiu ZR (2008) A rapid and efficient pre-deployment key scheme for secure data transmissions in sensor networks using lagrange interpolation polynomial. International Journal of Security and its Applications 2(3):49–55
15. Liu D, Ning P, Du W (2005) Group-based key pre-distribution in wireless sensor networks, Proceedings of 2005 ACM Workshop on Wireless Security, September 2, 2005, Cologne, Germany pp. 11–20
16. Huang D, Mehta M, Medhi D, Harn L (2004) Location-aware key management scheme for wireless sensor networks, Proceedings of the 2nd ACM Workshop on Security of Ad hoc and Sensor Networks, October 2004, Washington, DC, USA, pp. 29–42
17. Zhou L, Ni J, Ravishankar CV (2005) Efficient key establishment for group-based wireless sensor deployments. Proceedings of 2005 ACM Workshop on Wireless Security, Sept. 2005, Cologne, Germany pp. 1–10
18. Shi E, Perrig A (2004) Designing secure sensor networks. IEEE Wirel Commun 11(6):38–43. doi:10.1109/MWC.2004. 1368895
19. Unlu A, Levi A (2008) Two-tier, location-aware and highly resilient key predistribution scheme for wireless sensor networks, Proceedings of Visions of Computer Science—BCS International Academic Conference, London, UK, September 2008, pp. 355–366
20. Perrig A, Szewcyzk R, Wen V, Culler D, Tygar JD (2002) Spins: security protocols for sensor networks. Wirel Netw 8(5):521–534. doi:10.1023/A:1016598314198
21. Dressler F (2008) Authenticated Reliable and Semi-reliable Communication in Wireless Sensor Networks. International Journal of Network Security 7(1):61–68
22. Levi A, Tasci SE, Lee YJ, Lee YJ, Bayramoglu E, Ergun M (2009) Simple, extensible and flexible random key predistribution schemes for wireless sensor networks using reusable key pools. Journal of Intelligent Manufacturing, accepted for publication. http://dx.doi.org/10.1007/s10845-009-0256-z