Two-Tier, Location-Aware and Highly Resilient Key Predistribution Scheme for Wireless Sensor Networks[1]

Abdülhakim Ünlü, Albert Levi
Sabanci University, Faculty of Engineering and Natural Sciences
Orhanli, Tuzla, Istanbul 34956, Turkey
aunlu@su.sabanciuniv.edu, levi@sabanciuniv.edu

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. 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

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, 5, 6], symmetric cryptography is still more efficient to provide security in sensor networks. Symmetric cryptography necessitates pairwise keys distributed among the sensor nodes. The problem of distribution of keys to large number of sensor nodes is an active research area.

Key predistribution schemes [1, 7-12] 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, 16, 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 λ+1 or more nodes are captured all secure links are compromised.

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

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 has node-to-node authentication property. Keys and IDs of nodes are linked, so that nodes can verify the identity of each other.

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.

TABLE 1: Symbols and Parameters

N	number of nodes in each zone
Z	number of zones in the sensor network (= Z_x x 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 .. ω
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}

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 one used in [8]. In our deployment model, we divide the rectangular sensor field into a grid of $Z = Z_x$ x Z_y equal sized zones. Sensor nodes are grouped into Z equal sized groups each has N nodes. Centre 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. The nodes that belong to G_{ij} are dropped over deployment point d_{ij}. The actual location of sensor nodes after deployment is their resident points, 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. Using a Gaussian distribution, sensor nodes dropped at the same deployment point tend to be closer to each other.

2.2 Predistribution Phase

In key predistribution phase, we describe the method of how keys are distributed to nodes. We define two methods in this 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.

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.

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}, 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.

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. 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 $1 \leq m \leq Z$ and $1 \leq n \leq N$, 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 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 of queries goes on until either a hop-limit is reached or an agent node 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 Figure 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.

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.*

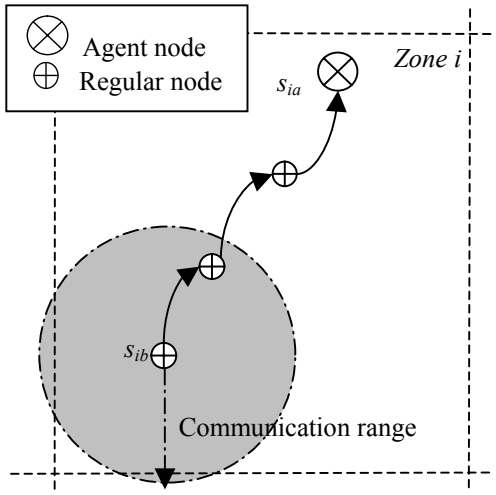

FIGURE 1: Regular node s_{ib} establishes a pairwise key with agent node s_{ia} using *hybrid key establishment method*

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.

3. PERFORMANCE EVALUATION

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 1000m x 1000m
- Deployment area is divided into 10 x 10 zones, i.e. $Z_x = Z_y = 10$ and $Z = 100$

- Total number of sensor nodes is 10000 and there are 100 nodes in each zone, i.e. N = 100.
- Communication range, R, for each node is 40m.

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 Figure 2, 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.

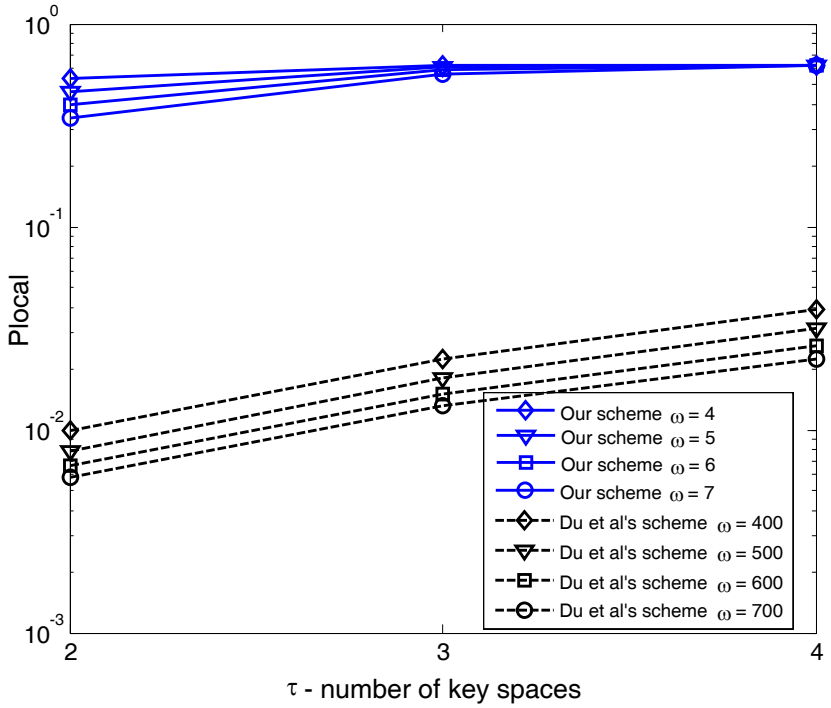

FIGURE 2: Local connectivity, P_{local}, vs. τ, number of key spaces installed in a node

Figure 3 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 Figure 3, we simulated our scheme for various values of λ+1 and τ values. We took 15, 25 and 35 as λ+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 Figure 3. When λ+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.

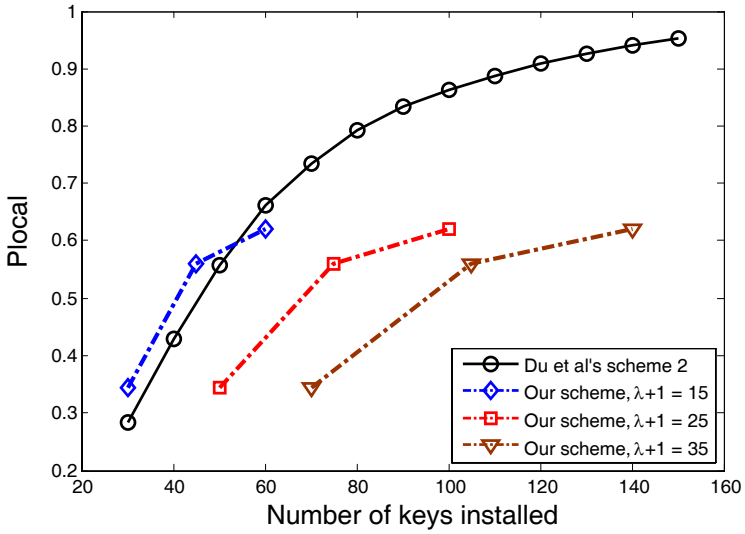

FIGURE 3: Local connectivity for Du et al.'s scheme 2 [8] and our scheme. For our scheme ω = 7, τ = 2, 3, 4, and λ+1=15, 25, 35. Number of keys in a sensor node is calculated as $\tau \times (\lambda+1)$

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 4 shows global connectivity of our scheme for τ=2, 3, 4 and ω=4, 5, 6. Simulation results indicate that even in the worst case where τ = 2 and ω=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.

We first determine average number of hops required to connect two neighboring nodes using intra-zone path key establishment. Figure 5 illustrates number of hops and connectivity values of corresponding secure link graphs for various τ and ω combinations. It can be observed from Figure 5 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

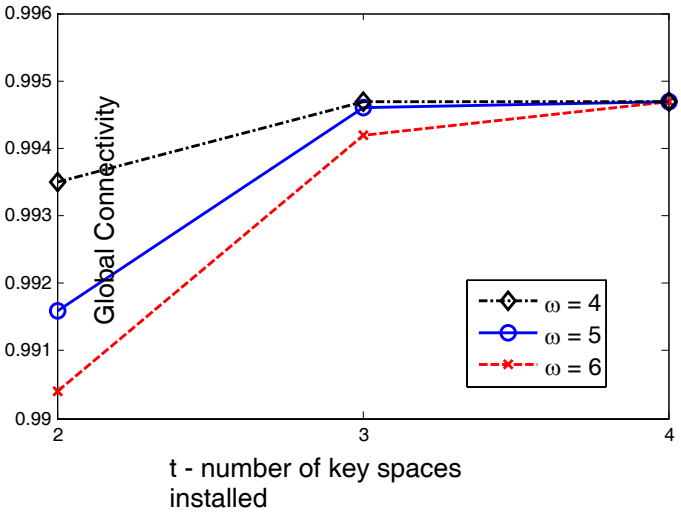

FIGURE 4: Global connectivity of our scheme

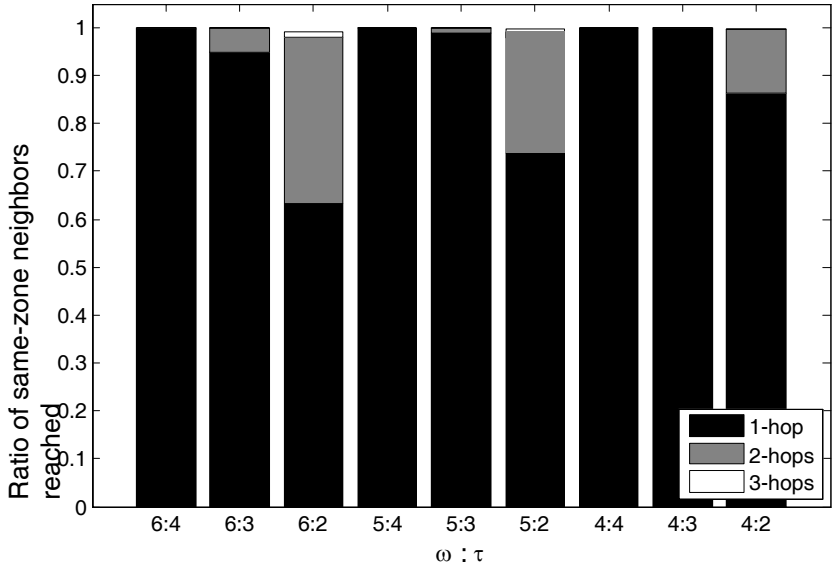

FIGURE 5: Communication overhead for intra-zone path key establishment in our scheme

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 Figure 6 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.

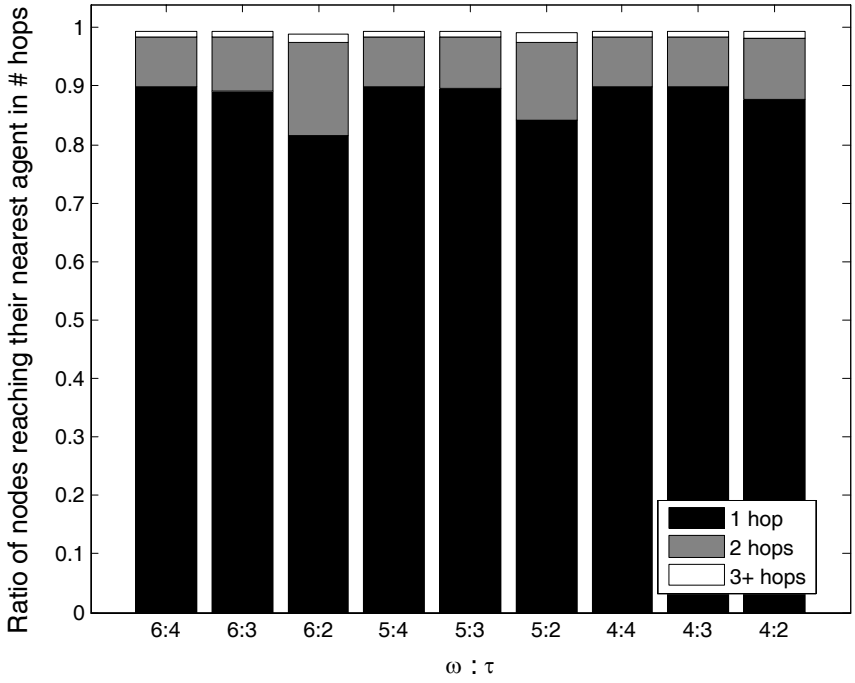

FIGURE 6: Ratio nodes reaching their nearest zone agent in *i* hops when A_z=10 (for our scheme)

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:

$$2(h_1 + h_2) + 3 \qquad [1]$$

We calculate the number of messages exchanged for each inter-zone path key. Figure 7 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.

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. 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].

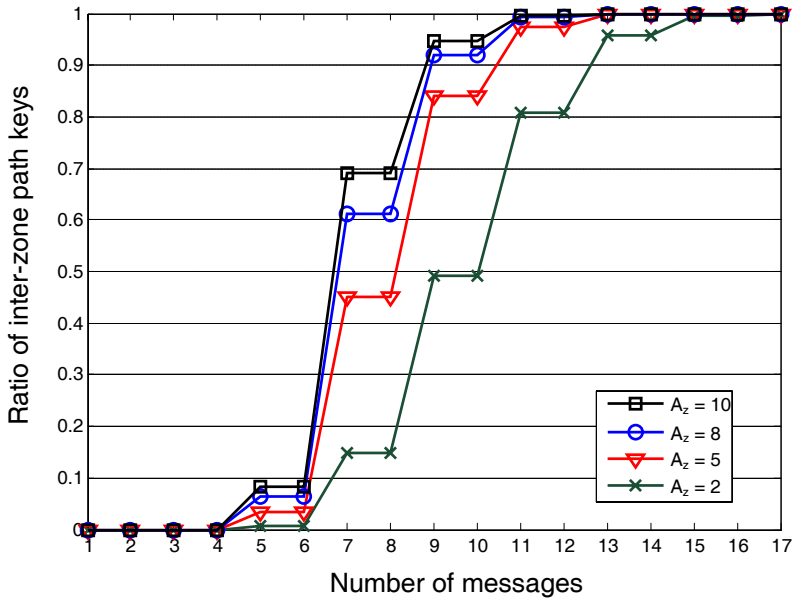

FIGURE 7: Communication cost for inter-zone path key establishment in our scheme, where ω=6 and τ =2

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 Figure 8, 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.

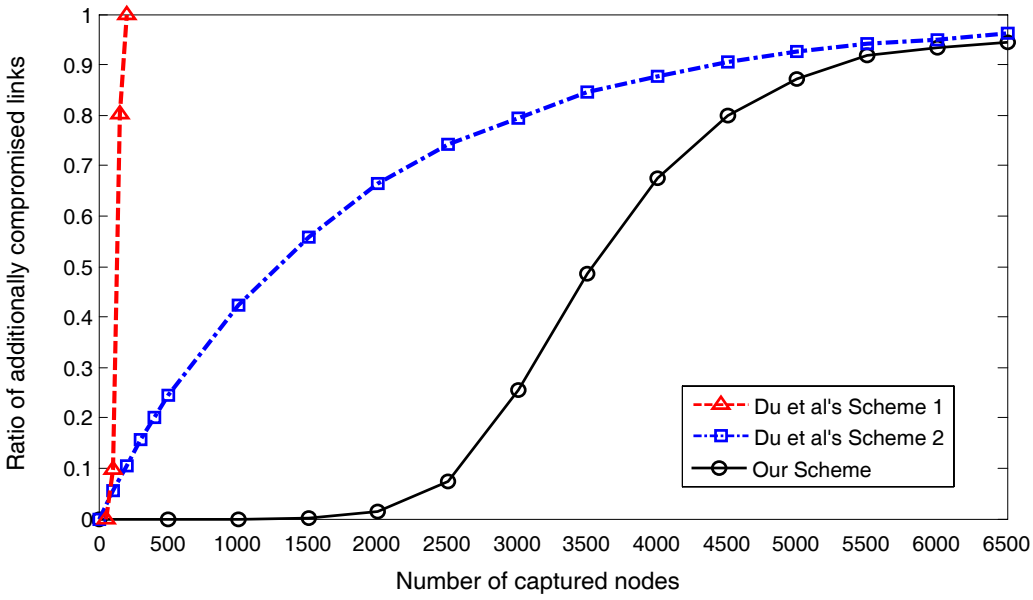

FIGURE 8: Ratio of additionally compromised links vs. number of nodes captured for our scheme, Du et al's scheme 2 [8] and Du et al's scheme [9]. P_{local} is approximately 0.56 for al schemes.

It can be observed from Figure 8 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 percent of secure links by capturing only 2000 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.

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 separate 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. We showed that by using a two-tier approach, our scheme achieves substantially strong node capture resiliency. Ratio of additionally compromised links when 2000 nodes (out of 10000 total nodes) are captured is almost zero.

REFERENCES

[1] Eschenauer, L. and V. D. Gligor. (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 vol. 209, pp. 335-338, Springer.

[3] Akyildiz, I. F., W. Su, Y. Sankarasubramaniam, and E. Cayirci. (2002) A survey on sensor networks. *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114.

[4] Malan, D. (2004) Crypto for Tiny Objects. Harvard University Technical Report TR-04-04.

[5] Gaubatz, G., J. Kaps, and B. Sunar. (2004) Public Keys Cryptography in Sensor Networks – Revisited. *Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS)*, LNCS vol. 3313, pp. 2-18, Springer.

[6] Watro, R., D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus. (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., A. Perrig, and D. Song. (2003) Random key predistribution schemes for sensor networks. *IEEE Symposium on Research in Security and Privacy*, pp. 197–213.

[8] Du, W, J. Deng, Y. S. Han, S. Chen, and P. Varshney. (2004) A key management scheme for wireless sensor networks using deployment knowledge. *Proceedings of IEEE INFOCOM'04*, March 2004.

[9] Du, W., J. Deng, Y. S. Han, and P. Varshney. (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. and P. Ning. (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. and P. Ning. (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., S. Setia, and S. Jajodia. (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. and M. Kuhn. (1996) Tamper Resistance – a cautionary note. *Proceedings of the Second Usenix Workshop on Electronic Commerce*, pp. 1-11.

[14] Spencer, J. (2000) *The Strange Logic of Random Graphs*, Algorithms and Combinatorics 22, Springer-Verlag, ISBN 3-540-41654-4.

[15] Liu, D., P. Ning, and W. Du. (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., M. Mehta, D. Medhi, and L. Harn. (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., J. Ni, and C. V. Ravishankar. (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. and A. Perrig. (2004) Designing Secure Sensor Networks, *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38- 43.