International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-1 Issue-5, October 2012

Location Dependent Key Management Scheme for Wireless Sensor Network

Kamal Kumar, Poonam Sharma

Abstract: ***WSN is an emerging technology and have great potential to be employed in critical situations like battlefields and commercial applications such as building, traffic surveillance, habitat monitoring and smart homes and many more scenarios. One of the major challenges wireless sensor networks face today is security. To deliver data without being compromised, WSN services rely on secure communication and key distribution. The existing key management solutions require a large memory space and more number of keys to be stored on each sensor node and still compromise of sensor nodes is critical issue, which must be considered. LDK is most widely used scheme now a days so we tried to work on this scheme. The aim of this paper is to analyze the proposed key management schemes for WSN based on their performance by realizing different environments. The analysis has been done theoretically and implemented through JAVA.***

Keywords: ***LDK, LEAP, SPINS, WSN.***

I. INTRODUCTION

A Wireless Sensor Network (WSN) [1] is a network composed of a large number of low-cost, low-power, multifunctional sensor nodes that are deployed for monitoring the physical world. Sensor nodes are also often called motes. The term originates from dust motes i.e., a single piece of dust. The main components of a sensor node are a micro-controller, memory, transceiver, power source and one or more sensors to perform measurements of some physical phenomena. The power source is mostly a battery. Sensor nodes are deployed in a sensor field. The deployment can be either done directly, by placing the sensor nodes in specific positions, or randomly, e.g., via aerial scattering in inaccessible terrains or disaster relief operations. Thus, the position of the sensor nodes in the Sensor field may not be known in advance. After deployment, the sensor nodes perform some self-organization mechanisms to set up the network, e.g., by determining their neighbors and setting up routing tables. Self-organization is also required to adapt to changes of the network, e.g., caused by node failure due to energy exhaustion. During operation of the WSN, sensor nodes perform measurements of some physical phenomena, e.g., the temperature at a certain location. This data is sent to one (or more) base station(s), called sink, for further processing. Since the transmission range of a sensor is limited, the sink may not be directly reached. Thus, messages are forwarded in a multi-hop communication to other sensor nodes which act as routers. Also sensor nodes may perform some operations on the data, e.g., data aggregation to decrease the amount of transmitted data. Since the transmission of data is much more cost-intensive than data processing, this is a commonly used approach to decrease the overall energy consumption. However, this may not be possible in all scenarios.

In LDK [4] the keys are allocated to sensor nodes depending on the location of the sensor node after deployment. This is done without requiring any knowledge about the deployment of sensors. This scheme starts off with loading a single key on each sensor node prior to deployment. The actual keys are then derived from this single key once the sensor nodes are deployed. LDK has low cost and hence the entire network is bootstrapped from a single common key loaded on all the sensors before deployment. LDK also allows for nodes to be added to the network anytime during the lifetime of the sensor network. As a result this approach not only reduces the number of keys that have to be stored on each sensor node but also provides for the containment of node compromise. Thus the compromise of a node in a location affects the communications only around that location. In addition, the proposed scheme is also low cost in that we start off with loading a single key on each sensor nodes are deployed.

II. KEY MANAGEMENT SCHEME

Key management is the set of techniques and procedures which support the establishment and maintenance of keying relationships between authorized parties, and covers the following:

- initialization of system users within a domain
- generation, distribution, and installation of keying material
- control over the use of keying material
- updating, revocation, and destruction of keying material
- store, backup/recovery, and archival of keying material.

A keying relationship is the state wherein nodes share common data (keying material) in cryptographic techniques [2]. This data may include public or private key pairs, secret keys, initialization values, and non-secret parameters. Authentication is the basis of secure communication, without a robust authentication mechanism in place, the remaining security goals (confidentiality, data integrity, and non-repudiation) are in most instances not achievable. Authentication can only be realized by means of verifying something known to be associated with an identity. Key management is concerned with the authenticity of the identities associated with the five procedures mentioned above. The fundamental function of key management schemes is the establishment of keying material, which in turn can be subdivided into agreement on a key and transport of this key.

Revised Manuscript Received on 30 October 2012
*****Correspondence Author(s)
Prof. Kamal Kumar*, Department of Computer Science and Engineering, Maharishi Markandeshwar University, Maharishi Markandeshwar Engineering College, Mullana- Ambala, India,
Poonam Sharma, Department of Computer Science and Engineering, Maharishi Markandeshwar University, Maharishi Markandeshwar Engineering College, Mullana-Ambala, India.

The WSN key management scheme consists of three main components:

- Key establishment
- Key refreshment
- Key revocation

Key establishment is about creating a session key between the parties that need to communicate securely with each other. Key refreshment prolongs the effective lifetime of a cryptographic key, whereas Key revocation ensures that an evicted node is no longer to able to decipher the sensitive messages that are transmitted in the network.

A. General Classification of Key Management Schemes

The classification of key management schemes on the basis of mechanism used in key establishment. In each mechanism, many schemes have been proposed to improve a basic scheme.

Pair-Wise Key Pre-distribution: A pair-wise key between a pair of nodes is directly stored, pre-distributed, in each node before node deployment (hereafter Pair-wise key scheme). Since each node in this scheme stores its pair-wise keys, it has perfect resilience against node capture which means even if a node is captured; the keys of non-captured nodes are never compromised [5]. However, scalability is limited because network scale depends on the memory of node where potential keys are stored.

Master Key Based Pre-distribution: A pair-wise key is derived from both a random number exchanged between each node and a single master key pre-distributed into each node (hereafter Master key scheme [6]). It results in great key connectivity and a little memory required. However, resilience is very low since all the pair-wise keys can be compromised when the master key is exposed to an adversary. Unlike Master key scheme which does not erase a master key after key setup, in 'LEAP' a master key is erased completely after a pair-wise key is established [7]. Although resilience is improved by erasing a master key of deployed nodes, there is still the risk of compromising a master key during node addition since added nodes store a master key.

Base Station Participation: 'SPINS' is included in this mechanism [8]. In SPINS, [3] each node is given its shared key with the base station. The base station directly transmits a pair-wise key respectively encrypted with each node's shared key. In other words, the base station intermediates in key setup. This scheme supports not only full connection but also perfect resilience. However, it is not scalable because of the terrible traffic volume resulting from intermediation.

Probabilistic Key Pre-distribution: For large networks, a probabilistic method is more efficient than a deterministic method. This mechanism results from the concept all the nodes in the entire networks are connected with the 0.9997 probability- almost fully connected- if the probability each node can establish a pair-wise key with its neighbor nodes is 0.33. A key ring is stored in each node before deployment (a key ring *k* is randomly selected from key pool *P* which is randomly selected from huge key space). A common key in both key rings of a pair of nodes is used as their pair-wise key. It guarantees enough resilience even though not perfect resilience, because the probability of breaking communication link is *k/P*. Moreover, it supports the large scale networks. The representative scheme is 'EG scheme'. Its variants are proposed like a combination of the EG scheme and the Blundo scheme and a combination of the EG scheme [9] and the Blom scheme [10, 11] which significantly enhance the security.

No Key Pre-distribution: This mechanism is considering the reality of wireless sensor networks. If an adversary does not know where and when nodes are deployed, it is difficult to launch active attack at an early phase. It can be a good trade-off to improve efficiency instead of a little node loss due to attacks during key setup. 'Key infection' is a representative scheme. In Key infection, key setup is completed in a relatively short time through a few transmissions [12]. The advantage in this mechanism is the base station does not take part in a key setup so that it consumes relatively less energy. Unlike the pre-distribution schemes above, it need not load potential keys into a node, which results in the low cost of network organization. However, it is only strong when an adversary does not observe communication during key setup and it cannot add nodes since a pair-wise key is established through exchanged data during key setup.

III. LOCATION DEPENDENT KEY MANAGEMENT SCHEME

The network scenario that we consider consists of resource constrained sensor nodes. Nodes can be added to this network at any point in time. The threat model that we consider assumes that the adversaries have very strong capabilities. The only constraint on their capability is that the adversaries will not be able to compromise a node for a small interval initially after the node is deployed. This interval can be of the order of milliseconds and definitely not more than a couple of seconds. After this initial interval an adversary might be able to compromise any node. Once the node is compromised, the adversary has access to all the keying material on the node. Following such a node compromise, the adversary is able to eavesdrop on all the links that have been secured using the compromised keying material.

Here we assume two types of nodes namely the regular sensor nodes as well as anchor nodes (AN). The only extra capability that an anchor node needs to have is the ability to transmit at different power levels. For example, WINS sensor nodes can transmit at 15 distinct power levels ranging from -9.3 to 15.6 dBm.

Here, we consider a network with Ns regular nodes and Na anchor nodes and three phases in the life time of these sensor nodes. These are

- Pre-deployment phase
- Initialization phase
- Communication phase

During the pre-deployment phase, every sensor node as well as every anchor node (AN) is loaded with a single common key K. We start with a single common key on all nodes in order to minimize the costs associated with key management. Following this the sensor and anchor nodes are deployed. The other two phases namely the initialization phase and the communication phase occur after deployment. During the initialization phase an AN transmits a beacon at each different power level. Each beacon contains a nonce (random number) encrypted using the common key K shared between all the nodes.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The beacons transmitted at different power levels contain a different set of nonce. Each sensor node receives a set of beacons based on the relative location of the sensor node and the various anchor nodes. The sensor node then decrypts each beacon message and obtains the nonce contained in each of the beacons. The sensor node then obtains the updated keys using a combination of the common key K and the received set of nonce. This procedure for obtaining updated keys is repeated by every sensor node in the network. As a result of this the keys on the various sensor nodes are location dependent. This is because sensors which are not in the same location receive a different set of nonce due to which the resulting keys are different.

Following the initialization phase, we have the communication phase. In this phase the sensor nodes set up secure links amongst themselves using the keys that they have from the initialization phase. In order to do this the set of neighboring sensor nodes will have to exchange messages that indicate the keys that they have. There are several possible ways to do this. In one case, a sensor node can select a number randomly and encrypt this number using all the derived keys that it has. Following this, the sensor node can transmit a message that consists of the random number (in plaintext) along with all the list of cipher-texts corresponding to this random number. A sensor node that receives such a message can encrypt the random number (in the received message) with its own cipher-texts. By comparing the received and computed list of cipher-texts, a sensor node can determine the number and value of the common derived keys. A pair of neighboring sensor nodes set up a secure link if the minimum number of common keys N_c that these pair of sensor nodes shares after the initialization phase is non-zero.

In this paper, we propose an approach for key management in sensor networks which takes the location of sensor nodes into consideration while deciding the keys to be deployed on each node. This approach which we call location dependent key management scheme not only reduces the number of keys that have to be stored on each sensor node but also provide containment of node compromise. The network scenario that we consider consists of 4 Anchor Nodes and 10 Regular nodes deployed uniformly in $(X-Y)$ co-ordinates of 50×50 area. We assume here that the position of ANs and RNs are fixed but the power level of ANs may vary.

Anchor node 1 is denoted as "$an11$" and its position is assumed at (0,0) in$(X-Y)$ co-ordinates.

Anchor node 2 is denoted as "$an21$" and its position is assumed at (50,0) in$(X-Y)$ co-ordinates.

Anchor node 3 is denoted as "$an31$" and its position is assumed at (50,50) in$(X-Y)$ co-ordinates.

Anchor node 4 is denoted as "$an41$" and its position is assumed at (0,50) in $(X-Y)$ co-ordinates.

Fixed Position of regular nodes as given below

Regular node 1 is denoted as "$n1$" and its position is assumed at (2,12) in $(X-Y)$ co-ordinates.

Regular node 2 is denoted as "$n2$" and its position is assumed at (10,7) in $(X-Y)$ co-ordinates.

Regular node 3 is denoted as "$n3$" and its position is assumed at (12,20) in $(X-Y)$ co-ordinates.

Regular node 4 is denoted as "$n4$" and its position is assumed at (17,2) in $(X-Y)$ co-ordinates.

Regular node 5 is denoted as "$n5$" and its position is assumed at (22,22) in $(X-Y)$ co-ordinates.

Regular node 6 is denoted as "$n6$" and its position is assumed at (27,27) in $(X-Y)$ co-ordinates.

Regular node 7 is denoted as "$n7$" and its position is assumed at (32,32) in $(X-Y)$ co-ordinates.

Regular node 8 is denoted as "$n8$" and its position is assumed at (37,37) in $(X-Y)$ co-ordinates.

Regular node 9 is denoted as "$n9$" and its position is assumed at (42,20) in $(X-Y)$ co-ordinates.

Regular node 10 is denoted as "$n10$" and its position is assumed at (47,47) in $(X-Y)$ co-ordinates.

IV. RESULT AND ANALYSIS

In this paper, we investigate the performance of scheme namely LDK and compare the performance of LDK with the basic random key pre-distribution scheme given in [13]. The implementation is done using JAVA. The following graphs show the results of analysis:

A. Connectivity ratio:

For a given node this is defined as the ratio of the number of neighbors of the node with which it can form secure links to the total number of neighbors of the node. The connectivity ratio for the network is then the average of the connectivity values for each of the nodes in the network. Note that this is a value between 0 and 1 with one which is the most desirable value indicating complete connectivity amongst every node and its neighbors. Figure shows the connectivity ratio of LDK. The connectivity ratio of LDK lies between 0.95-1.00 and it is constant for all nodes in the network. We also see from figure that the connectivity ratio is independent of the power level when the number of common key between the sensor nodes is one.

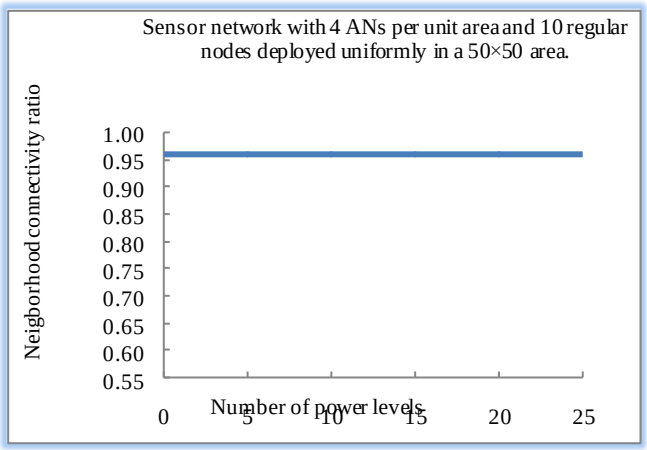

Fig. 1: Connectivity ratio between sensor nodes

B. Compromise ratio:

The compromise ratio is defined as the ratio of the number of secure links formed by the non-compromised nodes that have become vulnearble to the total number of secure links formed by non-compromised nodes in the network. The secure links become vulnerable on account of the leakage of keying material on the compromised nodes. Note that we donot consider the links formed by the compromised nodes.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The compromise ratio is zero when no node in the network is compromised. A good key management scheme should have a value of zero for the compromise ratio even when the network has compromised nodes. For example the pairwise key scheme will have a compromise ratio value of zero since compromise of any node does not impact the links formed by any of the non-compromised nodes. A value of 1 indicates that none of the links formed by the non-compromised nodes are secure after the compromise. For example, the single network wide key scheme will have a compromise ratio value of 1 as soon as at least one node is compromised. We assume that there are only 4 anchor nodes and 10 regular nodes in our network. The following table shows the percentage of secure links compromised with respect to the increasing power levels(upto 15).

Table I: Compromise Ratio

Number of power levels	Total Number of nodes having the same set of keys with any other node in the network	% of secure links Compromised
1	10	100
2	7	70
3	6	60
4	4	40
5	4	40
6	2	20
7	0	0
10	0	0
15	0	0

The graph shown in Fig. 2 is designed according to the data obtained in the Table I i.e the no. of compromised nodes decreases with respect to the increasing power levels(upto 10). The compromise ratio of LDK decreases as the power level increases.

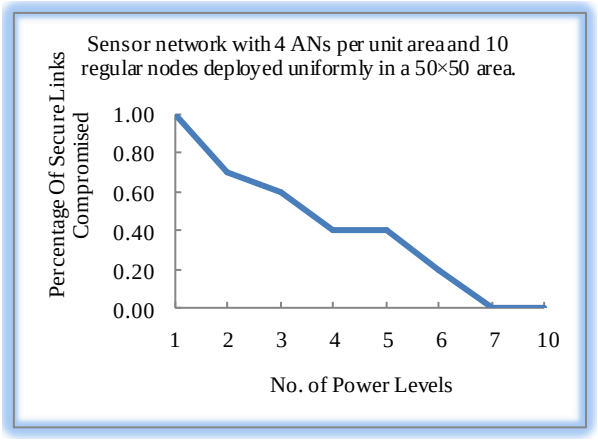

Fig. 2: Effects of security of non-compromised nodes

C. Number of hops required for communication:

The difference of the no. of hops required for communication in LDK and RKP is more visible from Fig. 3 where the number of hops required by location dependent key management scheme is very less as compared to the random key pre distribution scheme for communicating with their neighbors.

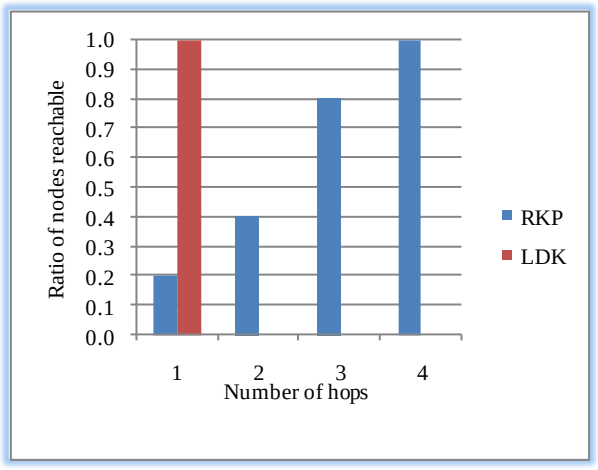

Fig. 3: Number of hops required for communication between RKP and LDK

V. CONCLUSION AND FUTURE WORK

In this paper, new approach has been proposed for sensor networks. The proposed scheme provides a different paradigm. The most important property of this scheme is that keys are allocated to sensor nodes depending on the location of the sensor node after deployment. This is done without the knowledge about the deployment of sensors. It also increases the security and scalability of the network. The performance is studied using the Java. The result shows, the scheme achieves the better security than random key distribution scheme while keep the merits of it. The analysis demonstrates a substantial improvement in compromise ratio.LDK also allows for nodes to be added to the network anytime during the lifetime of the sensor network. This scheme is also designed to be low cost and memory requirement is also very less.

There are some limitations still in LDK scheme, such as it consumes a large amount of energy on account of the great sized message transmission during pair wise key establishment. Besides, it provides opportunities for adversaries to launch attacks since key materials are exposed during exchange, so it need further study to improve this drawback so that it become more secure and energy efficient also.

REFERENCES

1. Ian Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. IEEE Communications Magazine 40, 8:102{114, 2002.
2. R. L. Rivest, A. Shamir, and L. M. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.
3. A. Perrig, R. Szewczyk, V. Wen, D. Cullar, and J. D. Tygar, “Spins: Security protocols for sensor networks,” in Proceedings of the 7th Annual ACM/IEEE Internation Conference on Mobile Computing and Networking (MobiCom), Rome, Italy, July 2001, pp. 189–199
4. F. Anjum, “Location dependent key management in sensor networks without using deployment knowledge,” in Communication Systems Software and Middleware, 2007. COMSWARE 2007. 2nd International Conference on, 2007, pp. 1–10.
5. Chan, H., Perrig, A., Song, D.: Random key predistribution schemes for sensor networks. In: Proceedings of the 2003 IEEE Symposium on Security and Privacy, pp.197-213(May-2003)

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

6. Lai, B., Kim, S., Verbauwhede, I.: Scalable session key construction protocol for wireless sensor networks. In: Proceedings of the IEEE Workshop on Large Scale RealTime and Embedded Systems LARTES (December 2002)
7. Zhu, S., Setia, S., Jajodia, S.: "LEAP: Efficient Security Mechanisms for Large- Scale Distributed Sensor Networks". In: Proceedings of the Tenth ACM conference on Computer and Communications Security, pp. 62–72 (October 2003)
8. Perrig, A., Szewczyk, R., Wen, V., Cullar, D., Tygar, J.D.: "SPINS: Security protocols for sensor networks". In: Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 189–199 (July 2001)
9. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: Proceedings of the 9th ACM conference on Computer and communications security, pp. 41–47 (November 2002)
10. Du, W., Deng, J., Han, Y.S., Varshney, P.K., Katz, J., Khalili, A.: A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. In: Proceedings of the ACM Transactions on Information and System Security, pp. 228–258 (August 2005)
11. Liu, D., Ning, P., Li, R.: Establishing Pairwise Keys in Distributed Sensor Networks. In: Proceedings of the ACM Transactions on Information and System Security, pp. 41–77 (February 2005)
12. Anderson, R., Chan, H., Perrig, A.: Key Infection : Smart Trust for Smart Dust. In: Proceedings of the 12th IEEE International Conference on Network Protocols, pp. 206–215 (October 2004)
13. L. Eschenauer and V. Gligor. A key-management scheme distributed sensor networks. In Proceedings of the 9th ACMConference on Computer and Communications Security, pages41-47, Nov 2002.

AUTHOR PROFILE

Prof. Kamal Kumar, received his M.Tech as well as B.Tech degree from Kurukshetra University Kurukshetra, India, Presently he is working as Associate Professor in Computer Engineering Department in M.M Engg College, Ambala, India. He is pursuing Ph.D from Thapar university, Patiala,India.

Poonam Sharma, received her B.Tech degree from Kurukshetra University Kurukshetra, India, and has done M.Tech in Computer Science and Engineering from Maharishi Markandeshwar University, Mullana-Ambala, Haryana,India.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)