Cluster-Based Key Management in Wireless Sensor Networks

Dr. Magdy E. Elhennawy

El-shorouk Academy, Institute of Computers and Information Technology
P. O. Box, Elshorouk, Cairo, Egypt
mhennawy@ad.gov.eg

Abstract: *Wireless sensor network (WSN) has unique nature which is different from other kinds of networks. It contains a large number of sensing devices which are limited in energy, computation power, and communication capabilities. WSNs are used in applications where the sensors have physical interactions with its environment and are accessible by any one. This makes them more vulnerable to various security threats. However, establishing secure communications is an open and difficult issue for WSNs. Exchanged information over WSN is to be secured by encryption and the use of an efficient key management protocol. Key management, in general, involves both the key establishment and key distribution. Traditional key management is difficult to apply in WSN due to its nature. In this paper, we present new protocol for key establishment and key distribution, referred to as cluster-based key management for WSNs, which is based on Eschenauer and Gligor (EG) key pre-distribution scheme. Our proposal covers the aspects of key generation phase using minimum number of keys generated in the pool, and re-keying phase using two key rings in each sensor node, to overcome the related drawbacks of the EG.*

Keywords: Wire less Sensor Networks, Security, Network Security.

1. Introduction

Key management is the process by which cryptographic keys are generated, protected, distributed, used, and finally destroyed. In WSNs this process consists of five phases which are: key generation, key pre-distribution (or simply, key distribution), shared key discovery, key establishment, and key update or re-keying phases. Key distribution is the process by which cryptographic keys are distributed through the network nodes. Due to the multi-hope communications, large number of nodes, resource constraints, and ad-hoc nature, the distribution of secret keys becomes one of the most challenging security components of WSN [1]. Further improvement for Eschenauer-Gligor scheme has been done in [12], the two nodes can establish a link only if they share q keys; in [15,17] the key pre-distribution scheme was developed and improved using multiple key spaces; and [9] described a random key pre-distribution scheme that uses deployment knowledge. In [18] the authors propose the random-pairwise keys scheme, which assures that, even when some numbers of nodes have been compromised, the remainder of the network remains fully secure.

In this paper an introduction is presented in this section, Main types of attacks and their consequences on the WSNs is described in section 2, protection mechanisms that can counter various attacks is described in section 3. As key management is a very important issue in encrypting information exchanged over networks, key management in WSNs has been is described in section 4. Various related WSNs schemes has been discussed in the section. The contribution of this paper has been presented in section 5. The simulation results has be discussed in section 5, then the conclusion has been presented in section 7.

2. Main Types of Attacks on WSN

Before a secure network can be realized, one must first identify the various types of network attacks and the potential security threats. However, the appropriate protection mechanisms can be properly defined. WSNs share the security threats of other communication networks, which are message interception, modification, and fabrication as well as interruption of communications and operation. The threats are specifically inherent in WSNs due to their wireless operation environment and their special constraints, which enable new forms and combinations of attacks. Even though WSNs have limited capabilities, an attacker can possess powerful tools, e.g. a laptop and a sensitive antenna, for making attacks more effective [1].

Attackers can be divided into outsiders and malicious insiders. Whereas an outsider is not an authorized participant of a WSN, an insider may have the knowledge of all the secret parameters of a WSN, such as cryptographic keys, and thus he is able to perform more severe attacks. An outsider can become an insider by compromising a WSN node.

Actually, sensors are mass-produced anonymous commodity devices that are initially unaware of their location. Once deployed, sensors should self-organize into a network that works unattended. Due to the fact that individual sensor nodes are anonymous and that communication among other sensors is via wireless links, sensor networks are highly vulnerable to other types of security attacks [2]. They can be spoofed or replayed routing attacks, selective forwarding attacks, sinkhole attacks, Sybil attacks, wormholes attacks, and Hello flood attacks [2, 3].

2.1 Spoofed or Replayed Routing

The most direct attack against a routing protocol is to target the routing information exchanged between nodes. By spoofing, altering, or replaying routing information, adversaries may be able to create routing loops, attract or repel network traffic, extend or shorten source routes, generate false error messages, partition the netork, increase end-to-end latency, etc [2].

2.2 Selective Forwarding

Selective forwarding is a way to influence the network traffic by believing that all the participating nodes in the network are reliable to forward the message. In selective forwarding attack, malicious nodes simply drop certain messages instead of forwarding every message. Once a malicious node picks on the messages, it reduces the latency and deceives the neighbouring nodes that they are on a shorter route. Effectiveness of this attack depends on two factors. First is the location of the malicious node. The closer it is to the base station, the more traffic it will attract. Second, the percentage of messages it drops. [3, 4].

2.3 Sinkhole Attack

In a sinkhole attack, the adversary's goal is to lure nearly all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole with the adversary at the center. Because nodes on, or near, the path that packets follow have many opportunities to tamper with application data, sinkhole attacks can enable many other attacks (selective forwarding, for example) [2].

In this attack, a malicious node acts as a black hole to attract all the traffic in the sensor network. Especially in a flooding based protocol, the attacker listens to requests for routes then replies to the target nodes that it contains the high quality or shortest path to the base station. Once the malicious node has been able to insert itself between the communicating nodes (for example, sink and sensor node), it is able to do anything with the packets passing between them. Figure (1) shows the conceptual view of a sinkhole attack.

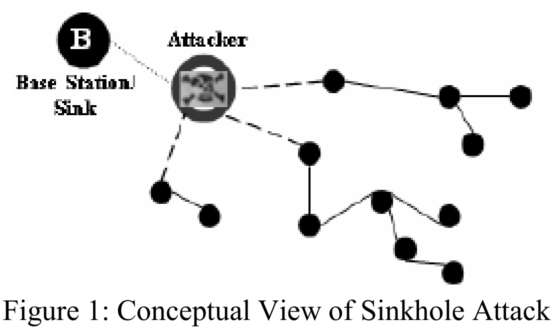

Figure 1: Conceptual View of Sinkhole Attack

2.4 Sybil Attack

Sybil attack [5] is defined as a malicious device (Sybil nodes) illegitimately taking on multiple identities. Sybil attack tries to degrade the integrity of data, security and resource utilization that the distributed algorithm attempts to achieve. Sybil attack can be performed for attacking the distributed storage, routing mechanism, data aggregation, voting, fair resource allocation and misbehaviour detection. Basically, any peer-to-peer network (especially wireless ad hoc networks) is vulnerable to Sybil attack. However, as WSNs can have some sort of base stations or gateways, this attack could be prevented using efficient protocols. In [5] the authors develop taxonomy of Sybil attacks different forms. An approach to defend against Sybil attack is resource testing [2]. This assumes that each physical entity is limited in some resource. Verification of identity involves testing that each identity has as much of the tested resource as expected from each physical device. Some of the possible resources here are computation, storage, and communication.

2.5 Wormholes Attacks

In the wormhole attack, a malicious node tunnels messages received in one part of the network over a low latency link and replays them in a different part. Due to the nature of wireless transmission, the attacker can create a wormhole even for packets not addressed to itself, since it can overhear them in wireless transmission and tunnel them to the colluding attacker at the opposite end of the wormhole. The tunnel can be established in many different ways, such as through an out-of band hidden channel (e.g., a wired link), packet encapsulation, or high powered transmission. The tunnel creates the illusion that the two end points are very close to each other, by making tunneled packets arrive either sooner or with lesser number of hops compared to the packets sent over normal routes. This allows an attacker to subvert the correct operation of the routing protocol, by controlling numerous routes in the network. Later, he can use this to perform traffic analysis or selectively drop data traffic. The wormhole attack mainly consists in network layer attacks when the attack is classified according to network protocol stacks [6].

Wormholes can exploit routing race conditions which happen when node takes routing decisions based on the first route advertisement. Attacker may influence network topology by delivering routing information to the nodes before it would really reach them by multi hop routing [5]. Figure (2) shows a situation where a wormhole attack takes place.

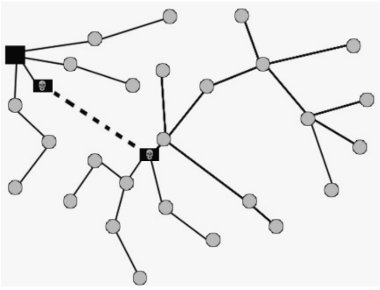

Figure 2: Wormhole Attack

2.6 Hello Flood Attack

This attack uses Hello packets as a weapon to influence the sensors in WSN. Many WSN routing protocols require nodes to broadcast Hello packets after deployment, which is a sort of neighbour discovery based on radio range of the node, Laptop class attacker can broadcast Hello message to nodes and then advertises high-quality route to sink [5]. This is illustrated in Figure (3).

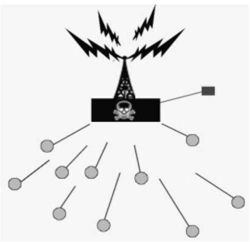

Figure 3: HELLO flood attack

3. Protection Mechanisms

To counter the network attacks one or more protection mechanisms should be used. The protection mechanism is designed to detect, prevent, or recover from a network attack. Network protection mechanisms enhance the network security and eventually information transfer. One of the basic protection services upon which computer networks securities are based can be confidentiality. Encryption, as a protection mechanism, can implement the confidentiality protection service to protect WSN. Accordingly, the other basic security protection mechanism that can be applied is the key management.

However, to achieve security in WSN, it is important to be able to encrypt messages sent among sensor nodes. Keys for encryption purposes must be shared upon by communicating nodes. Due to resource constraints, achieving such key agreement is nontrivial. Many key management schemes used in general networks, such as Diffie-Hellman and public-key-based schemes, are not suitable for wireless sensor networks [7].

4. Key Management in Wireless Sensor Networks

To encrypt messages exchanged over sensor nodes properly, keys for encryption must be shared with by the communicating nodes. WSN, on the other hand, need to employ key management protocols that scale to a large number of sensor nodes with limited capabilities [7, 8].

4.1 Security Constrains in Wireless Sensor Networks

To apply key distribution in wireless sensor networks, we need to consider factors such
as [9, 10, 11]: A) Limited power, which limits the life time of keys. Once sensor nodes are deployed in the field, they cannot be recharged. On the other hand, battery replacement might cause the device to reinitialize and zeros out keys. B) The network has very large number of nodes. C) Limited computation capabilities in nodes makes available processing power very limited. D) Working memory of sensor nodes is insufficient to even hold variables for asymmetric cryptographic algorithms. E) Inability to predetermine the neighbors after deployment as well as inability to put absolute trust in neighbors.

WSNs are dynamic in the sense that they allow addition and deletion of sensor nodes after deployment to grow the network or to replace weak spot and unreliable nodes. WSNs may be deployed in hostile areas where communication is monitored and nodes are subjected to capture and covert use by an adversary [12]. Therefore, solution must compromise between the above factors [1, 9, 11].

4.2 Key Distribution Schemes

There are three schemes that are used in WSN for key distribution; each has its advantages and disadvantages. They are network keying, Pairwise keying, and group keying [9, 12, 13]. Network keying is used to secure broadcast messages. The only disadvantage is the lacking of robustness; while it is characterized by simplicity, allowing data aggregation and fusion, scalability, self-organizing, and flexibility. Pairwise keying is used to secure unicast communication between a pair of sensor nodes over single or multi-hop wireless link. It has many advantages such as its robustness and authentication for each node. On the other hand, it is non-scalable, and unable to self-organize and inflexible one [7, 14]. Group Keying is used to secure multicast communication among a group of sensor nodes over single or multi-hop wireless link. It allows multicast, group collaboration, and presents better robustness over network keying. It has adjustable scalability, flexibility, and able to be self-organize within cluster. Although all these benefits, it still has many disadvantages which are: lacking for efficient storage for group keying in IEEE

802.15.4, difficult to set up securely, and its Cluster formation information is application dependent [7, 14].

4.3 Key Agreement Schemes

A key agreement scheme is a self-enforcing scheme, in which each node takes role in establishing a common secret through mutual exchanged messages between the nodes in a protected method. Although these protocols are nearly fully distributed, they are unpractical in WSNs for many reasons. First, these protocols are not strong enough for variable topology or repeatedly irregular links, which takes place in WSNs. To create an unbeaten shared key, the invariant network topology, with the result that a routing infrastructure is essential because pairwise nodes may not reach each other and need to intermediate nodes to convey the messages. In addition, all nodes need to be online before the key agreement process is over. If a node leaves in the midst due to link or battery outage, the remaining nodes need to rerun the process from scratch. Second, a key agreement scheme that depends on asymmetric cryptography is not computationally efficient. Finally, due to the frequent interactive re-keying, the scalability in a key agreement scheme may be troublesome [9,11,15]. A. Perrig et al. [17] introduces Security Protocols for Sensor Networks (SPINS). In SPINS, each sensor node shares a secret key with the base station. To establish a new key, two nodes use the base station as a trusted third party to set up the new key.

4.4 Key Pre-distribution Schemes

Key pre-distribution schemes have been proposed as tools to overcome wireless sensor network constraints such as limited communication and processing power. Two sensor nodes can establish a secure link with some probability based on the information stored in their memories, though it is not always possible that two sensor nodes may set up a secure link [12].

The pure Random Key Pre-distribution scheme (R-KPS) was first proposed by Eschenauer and Gilgor (EG Scheme) [16] where each sensor node receives a random subset of keys from a large key pool before deployment. To agree on a key for communication, two nodes find one common key within their subsets and use that key as their shared key. It is based on probabilistic key sharing and a simple shared-key discovery protocol for key distribution, key revocation, and node re-keying [15].

EG random key pre-distribution scheme (R-KPS) works as follows: before sensor nodes are deployed, an initialization phase is performed. In the initialization phase, EG scheme picks a random pool (set) of keys S out of the total possible key space. For each node, m keys are randomly selected from the key pool S and stored into the node’s memory. This set of m keys is called the node’s key ring. Upon deployment, any pair of nodes can establish a secure and direct connection if they share at least one common key. Because the keys are randomly distributed to each sensor node, not every pair of nodes has common keys. Hence, any two nodes without common keys are required to establish an indirect connection via an intermediary node that shares a common key with both nodes. More specifically, R-KPS consists of three different phases: key pre-distribution, shared key discovery, and path-key establishment.

Key pre-distribution phase takes place before the sensor nodes are deployed. During this phase, a large pool of random keys and their key identifiers are generated. A subset of randomly chosen keys and their associated key identifiers are assigned to each sensor. This phase is to guarantee that, with a small number of keys, a common key is established with high probability between two or more sensors during the shared key discovery phase.

The shared-key discovery phase performs during WSN initialization. Each node discovers its neighbors with which it shares keys within the wireless communication range. A simple way for any two nodes to discover if they share a key is that each node broadcasts the list of identifiers of the keys in their key ring in clear text. A more secure method is that, for each key on a key ring, each node broadcasts the string:

$\{\alpha, E_{Ki}(\alpha), i=1, 2, .., n\}_,$ where α is a challenge.

Once $E_K(\alpha)$, is decrypted with a proper key by a recipient, the challenge is revealed and thus a common key shared between the broadcasting node and the recipient is established. This phase can establish a direct secure link between two nodes if they share a key.

The path-key discovery phase executes after the shared-key discovery phase. In this phase, a path-key is established for any selected sensor pairs that do not share a common key but can be connected by two or more direct links created in the shared-key discovery phase. If the key graph is connected, an indirect link can be found from a source node to the target node. The source node can then generate a path-key and send it securely via the path to the target node.

In the EG scheme, a vital issue is to pick the right parameters, such that the graph generated during the key-setup phase (or path-key discovery) is connected.

A fundamental problem regarding securing WSNs is to choose the relevant parameters, pool size, and key ring size, to achieve a very high probability of connectivity, refer to formulas in[16]..

5. Proposed Scheme

To counter the above stated WSNs attacks, there is a need to employ encryption algorithms to implement confidentiality security service. According to the nature of WSN, symmetric-key is appropriate to se, leading to the need to a more efficient key management scheme. As previously stated, the key management for wireless sensor networks includes six phases which are: key generation, key pre-distribution (or simply, key distribution), shared key discovery, key establishment, and key update or re-keying phases. Pre-distribution of secret keys is possibly the most practical approach for securing wireless sensor network communications. The proposed scheme focuses on key pre-distribution schemes, which are appropriate for WSN. The proposed scheme is based on Eschenauer and Gligor (EG) key pre-distribution scheme [16] and improves the re-keying phase and optimize the number of generated keys.

5.1 Key Generation and Pre-distribution Phase

We have developed our own simulation model and tested the outputs using MATLAB 7.0.1. It runs over PC environment. The design of the simulation model, to explore the proposed scheme, depends on the generation of a pool of P keys and keys identifiers, picking randomly two k out of P keys without replacement to establish the key rings of a sensor, loading the two key rings into the memory of each sensor; saving the keys identifiers of a key ring and associated sensor identifier on the base station; and for each node, loading the i-th base station with the key shared with that node. The base station has the right to change between the two key rings of the sensor through implicit command when one of them is compromised and need to be replaced.

Hereby, we use the negative selection algorithm [19] in which the system is trained using the normal messages and then produces antibodies for the defected messages. After generating two random key rings from the key pool for each node, the Negative Selection Algorithm (NSA) will be applied on those two key rings, to make sure there aren't any similarities. Then, loading the two key rings into the memory of sensor node. Another difference is that the size of the key pool is less than the one of the EG scheme.

To calculate the critical parameter for the size of the key pool $|\mathrm{P}|$ in our proposal, we use the protocol described in [20]. If the key pool size is too large, then the probability of any two nodes sharing at least *q* keys would be less than *p*, and the network may not be connected after bootstrapping is complete. If the key pool size is too small, then we are unnecessarily sacrificing security.

$p_{connect}$ is the probability of any two nodes sharing sufficient keys to form a secure connection.

$p_{connect}$=1- (probability that the two nodes share insufficient keys to form a connection)

$p_{connect} = 1 - (p(0) + p(1) + \ldots\ldots\ldots + p(q-1))$

For a given key ring size m, minimum key overlap *q*, and minimum connection probability *p*, we choose the largest $|\mathrm{P}|$ such that $p_{connect} \geq p$.

5.2 Re-keying Phase

Whenever a sensor node is compromised inside the cluster, the cluster head will take an action to switch over between two pre-loaded key rings, so the key rings will be changed in all sensor nodes. And, both the shared key discovery phase, and the path key establish phase will be repeated as in EG scheme.

When the compromise is repeated for the second time, the cluster head will send an order to the sensor nodes to apply a predefined respond function that performs some mathematical operations (transformation) between the two pre-loaded key ring to generate new key ring. Then revoke the oldest key ring to be replaced by the new one, which will be the main key ring inside the sensor node. In our experiment we used predefined respond function as a subtraction function to produce a new key ring. The subtraction function algorithm pseudo code can be as follows:

```
Main ()
{  initialize the contents of the 1st  sinsor ring
   initialize the contents of the 2nd  sinsor ring
   for every row i
   {  do   for every column j
          { do
                  {  set the value at i , j of the output  matrix to the difference between the
                     element at i, j of the first matrix and the element at i , j of second matrix
                  }
            done
          }
      Done
   }
}
```

6. Simulation Results

Our scheme proposes the following: 1) Number of keys in the pool is minimized to the value that allows optimum key share between key rings stored, 2) Key pre-distribution of each one node with two key rings, and 3) Applying the concept of clustering to allow grouping of sensor nodes.

First, the minimized number of keys in the pool to the value that allows optimum key share between key rings stored, will increase the probability of the key share between rings in various sensor nodes. Second, the key pre-distribution of each one node with two key rings will allow switching from the first to the second ring automatically, in the one ring, as needed. For example when one ring is compromised. This process is accompanied with, a concurrently, regeneration of a new ring, according to a pre-defined agreed upon pulse from the base sensor. This allows the renewal of the key rings in the sensor nodes without the need to any communications for re-keying. Third, applying the concept of clustering to allow grouping of sensor nodes (may allow manipulation of heterogeneous WSNs in one WSN), simplifying the process of rekeying process. Noting that we need to a inter-cluster management protocol.

On the other hand, in our simulation model, the matching analysis between the compromised key ring and the new generated one was done by using the negative selection algorithm. The key ring can be changed when matching occurs with certain threshold (affinity measure). Finally, the system of sensors, cluster, and base station are constructed with the final key ring.

Comparing the result of our proposed scheme with EG scheme using the EG conception of the computation formulas , Figure (4) illustrates the enhancement done by our proposal. The probability of generating smaller numbers of the keys are used to secure links is smaller with approximately 10%. This is due to using less number of keys in the pool and using the negative selection algorithm when we generate the key rings.

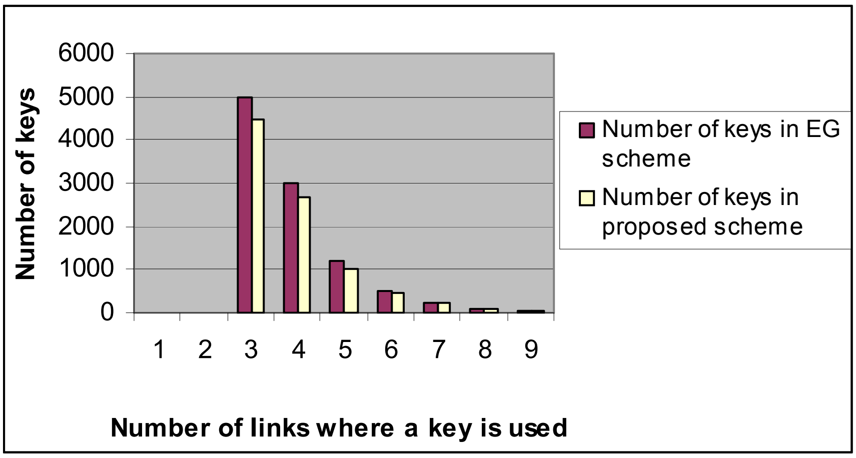

Figure 4: Comparing the number of keys between EG scheme and the proposed scheme

Figure (5) illustrates the enhancement done by our proposal, and depicts the probability of sharing at least one key versus the key ring size. We compare our results with EG scheme. The figure indicates that our scheme considerably improves local connectivity The local connectivity of the proposed scheme is better than that of EG scheme due to using the negative selection algorithm while creating the key rings.

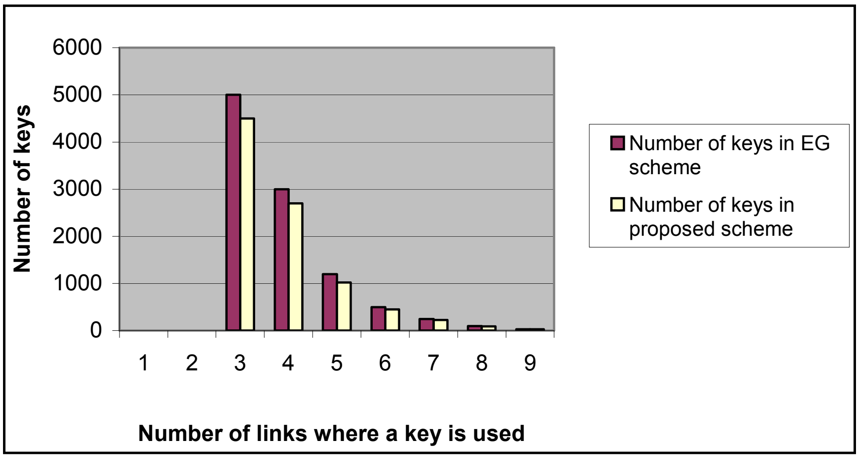

Figure 5: the Enhancement done by our Proposal

7. Conclusions :

When a sensor node is compromised the key rings will be switched over in all sensor nodes through a command from the cluster head. The system applies a predefined respond function that performs some transformation between the two pre-loaded key rings to generate new key ring, in case it compromise. So, the proposed scheme requires low communication overhead requirements, which make it suitable for wireless sensor network nodes. On the other hand it requires more storage requirements. Since each bit transmitted in WSNs consumes about as much power as executing 800-1000 instructions [21], thus communication is more costly than computation in WSNs. One of the important results of our contribution is the large reduction of consumed energy due to the reduction of the communication needed for re-keying, and also due to low communication overhead requirements.

Referances

[1] M. Kuorilehto, M. Kohvakka, J. Suhonen, P. Hamalainen, M. Hannik¨ainen, and Timo D. Hamalainen, "Ultra-Low Energy Wireless Sensor Networks in Practice Theory", Realization and Deployment, John Wiley & Sons, Inc., Publication, 2007.

[2] Chris Karlof, David Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures", Ad Hoc Networks 1, pp 293-315, 2003.

[3] M. Kuorilehto, M. Kohvakka, J. Suhonen, P. Hamalainen, M. Hannik¨ainen, and Timo D. Hamalainen, "Ultra-Low Energy Wireless Sensor Networks in Practice Theory", Realization and Deployment, John Wiley & Sons, Inc., Publication, 2007.

[4] Albert Y. Zomaya, "Algorithms And Protocols For Wireless Sensor Networks", A John Wiley & Sons, Inc., Publication, 2008.

[5] Newsome, J., Shi, E., Song, D, and Perrig, A, “The sybil attack in sensor networks: analysis & defenses”, Proc. of the third international symposium on Information processing in sensor networks, pp. 259 – 268ACM, 2004.

[6] Zaw Tun and Aung Htein Maw, "Wormhole Attack Detection in Wireless Sensor Networks", PWASET, Vol. 36, pp 549-554, DEC. 2008.

[7] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge", IEEE Infocom 2004.

[8] M. Eltoweissy, M. Moharrum and R, Mukkamala, "Dynamic Key Management in Sensor Networks", IEEE Transactions On Dependable And Secure Computing, Vol. 3, No. 1, January-March 2006.

[9] Xiao, "Security in Sensor Networks", Taylor& Francis Group, LLC, 2007.

[10] N. Bulusu, "Wireless Sensor Networks", Artech House, Inc, 2005.

[11] R. Shorey, A. Ananda, M. Choon Chan, and Wei Tsang Ooi, "Mobile, Wireless, And Sensor Networks Technology, Applications, and Future Directions", John Wiley& Sons, Ltd, 2006.

[12] M. Mehta, D. Huang, and L. Harn, "A Practical Scheme for Random Key Pre-distribution Scheme and Shared-Key Discovery in Sensor Networks", Proc. of 24th IEEE International Conf. on Performance Computing and Communication, 2004.

[13] Huang, D. Mehta, M., Medhi, D., and Harn, L., "Location-aware key management scheme for wireless sensor networks", 2nd ACM workshop on Security of Ad Hoc and Sensor Networks 2004.

[14] K. Sohraby, D. Minoli, and T. Znati, "Wireless Sensor Networks Technology, Protocols, and Applications", Fourth Edition, John Wiley& Sons. Inc, 2007.

[15] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks", Proc. 10th ACM Conf. Comp. and Commun. Security, pp.42-51, Oct 2003.

[16] L. Eschenauer and V. D. Gligor, “A key management scheme for distributed sensor networks", Proc. of the 9th ACM Conference on Computer and Communication Security, pp.41–47, Washington, D.C., USA, 2002.

[17] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A Key Pre-distribution Scheme for Sensor Networks Using Deployment Knowledge ", IEEE Transactions On Dependable And Secure Computing, Vol. 3, No. 1, pp.62-77, January-March 2006.

[18] Adrian Perrig, Haowen Chan, Dawn Song, "Random Key Predistribution Schemes for Sensor Networks" IEEE Symposium on Security and Privacy, 2003.

[19] Dipankar Dasgupta , "Advances in Artificial Immune Systems", IEEE Computational Intelligence Magazine, pp. 40- 49 Novembre 2006.

[20] Adrian Perrig, Haowen Chan, Dawn Song, "Random Key Predistribution Schemes for Sensor Networks" IEEE Symposium on Security and Privacy, 2003.

[21] Yong Wang Attebury, G. Ramamurthy, B., "A survey of security issues in wireless sensor networks", Vol 8, Issue 2, pp.2-23 Second Quarter 2006