# LSec: Lightweight Security Protocol for Distributed Wireless Sensor Network

Riaz Ahmed Shaikh, Sungyoung Lee, Mohammad A. U. Khan, and Young Jae Song

Department of Computer Engineering, Kyung Hee University,
Sochen-ri, Giheung-eup, Yongin-si, Gyeonggi-do, 449-701, South Korea
{riaz, sylee, khan}@oslab.khu.ac.kr, yjsong@khu.ac.kr

**Abstract.** Constraint specific wireless sensor networks need energy efficient and secure communication mechanisms. In this paper we propose Lightweight Security protocol (LSec) that fulfils both requirements. LSec provides authentication and authorization of sensor nodes with simple secure key exchange scheme. It also provides confidentiality of data and protection mechanism against intrusions and anomalies. LSec is memory efficient that requires 72 bytes of memory storage for keys. It only introduces 74.125 bytes of transmission and reception cost per connection.

## 1. Introduction

Wireless sensor networks consist of a large number of small size sensor nodes deployed in the observed environment. Sensor nodes have smaller memory (8K of total memory and disk space) and limited computation power (8-bit, 4 MHz CPU) [1]. They usually communicate with a powerful base station which connects sensor nodes with external networks. The limited energy at senor nodes creates hindrances in implementing complex security schemes. There are two major factors for energy consumption:

1. Transmission and reception of data.
2. Processing of query request.

Wireless networks are relatively more vulnerable to security attacks than wired networks due to the broadcast nature of communication [1]. In order to implement security mechanism in sensor networks, we need to ensure that communication overhead is less and consumes less computation power. With these constraints it is impractical to use traditional security algorithms and mechanism meant for powerful workstations.

Sensor networks are vulnerable to a variety of security threats such as DoS, eavesdropping, message replay, message modification, malicious code, etc. In order to secure sensor networks against these attacks, we need to implement message

This work is financially supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor (MOLAB) through the fostering project of the Lab of Excellency. The corresponding author of this paper is Prof. Sungyoung Lee.

confidentiality, authentication, message integrity, intrusion detection and some other security mechanism. Encrypting communication between sensor nodes can partially solve the problems but it requires a robust key exchange and distribution scheme.

In general, there are three types of key management schemes [2,3]: Trusted Server scheme, self enforcing scheme and key-predistribution scheme. Trusted server schemes relies on a trusted base station, that is responsible for establishing the key agreement between two communicating nodes as described in [4]. It uses symmetric key cryptography for data encryption. The main advantages of this scheme are, it is memory efficient, nodes only need to store single secret key and it is resilient to node capture. But the drawback of this scheme is that it is energy expensive, it requires extra routing overhead in the sense that each node need to communicate with base station several times [3]. Self enforcing schemes use public key cryptography for communication between sensor nodes. This scheme is perfectly resilient against node capture and it is fully scalable and memory efficient. But the problem with the traditional public keys cryptography schemes such as DSA [5] or RSA [6] is the fact that they require complex and intensive computations which is not possible to perform by sensor node having limited computation power. Some researchers [7,8] uses Elliptic curve cryptography as an alternative to traditional public key systems but still not perfect for sensor networks. Third scheme is key pre-distribution scheme based on symmetric key cryptography, in which limited numbers of keys are stored on each sensor node prior to their deployment. This scheme is easy to implement and does not introduce any additional routing overhead for key exchange. The degree of resiliency of node capture is dependent on the pre-distribution scheme [3].

Quite recently some security solutions have been proposed in [9,10,11,12,13] especially for wireless sensor networks but each suffers from various limitations such as higher memory and power consumptions that are discussed in section 4.

Keeping all these factors in mind we propose a lightweight security protocol (LSec) for wireless sensor networks. LSec combines the features of trusted server scheme and Self Enforcing security schemes. Our main contribution is the designing and implementation of LSec that provides

- Authentication and Authorization of sensor node.
- Simple Secure key exchange scheme.
- Secure defense mechanism against anomalies and intrusions.
- Confidentiality of data.
- Usage of both symmetric and asymmetric schemes.

The rest of the paper is organized as follows. Section 2 describes the details of LSec. Section 3 presents the simulation results and evaluation of LSec. Section 4 presents the comparison of LSec with other security solutions and Section 5 consists of conclusion and future direction.

## 2. Light weight Security Protocol (LSec)

The basic objective of LSec is to provide lightweight security solution for wireless sensor networks where all nodes can communicate with each other. LSec can support both static and mobile environment, which may contain single and multiple Base

Stations (BS). Basic system architecture is shown in figure 1. LSec uses both symmetric and asymmetric schemes for providing secure communication in wireless sensor networks.

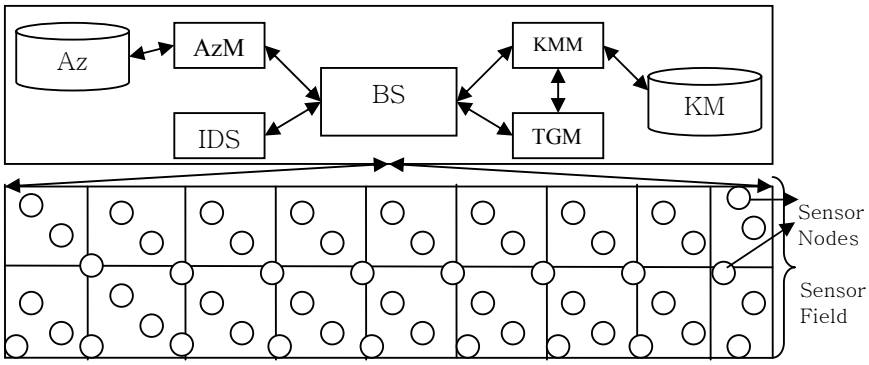


**Fig 1.** LSec System Architecture

Key Management Module (KMM) is used to store public and shared secret key of each node with BS to the database. Token Generator Module (TGM) is used to generate the tokens for the requesters, which will be further used by the other communicating party for the authentication of requester node. Authorization Module (AzM) is used to check whether a particular node is allowed to communicate with other node or group. Lightweight mobile agents will only be installed on Cluster heads which sends alerts messages to intrusion detection system (IDS), which is responsible for detecting any anomaly or intrusion in the network. Basic assumptions and rules of LSec are given below.

### 2.1 Assumptions

1. Base Station (BS) is the trusted party and it will never be compromised. Compromising the Base station can render the entire sensor network useless, and it is the only point from where sensor node can communicate with external networks.
2. Only Base Station (BS) knows the Public keys (Pk) of all the sensor nodes in the network. Communicating nodes will know each other's public key during the time of connection establishment.

### 2.2 Rules

- Asymmetric scheme will only be used for sharing ephemeral secret key between communicating nodes.
- For every session new random secret key will be used.
- Data will be encrypted by using symmetric schemes because these schemes

are considered to be executed three to four times faster than asymmetric schemes [14].

### 2.3 LSec Packet Format

LSec packet format is shown in table 1. Currently LSec uses seven types of packets, 'Request', 'Response', 'Init', 'Ack', 'Data', 'Update Group Key' and 'Alert' packet. All seven packets are distinguished by 'type' field in the LSec packet. IDsrc field contain the id of sending node and last encrypted portion contain the information depending upon the type of packet, as shown in table 1.

**Table 1.** LSec: Type field

| Type | $ID_{src}$ | Encrypted Portion |
|---|---|---|
| Request | Any (sensor node) | $EK_{A\text{-}BS}$ (Intended-$ID_{dest}$ , N) |
| Response | BS | $EK_{A\text{-}BS}$ (R-type, Intended-$ID_{dest}$ , N , Pk, token \| R) |
| Init | Any (sensor node) | $EK_B^+$(N, Pk, token) |
| Ack | Any (sensor node) | $EK_A^+$(N,sk) |
| Data | Any (sensor node) | EKsk (data) |
| UpdateGroupKey | Any CH sensor node | $EK_G$ (GroupID, new Key), MAC |
| Alert | Any CH sensor node | $EK_{CH\text{-}BS}$ (Alert-type), MAC |

$EK_{A\text{-}BS}$ = Encrypt with the secret key shared between node A and BS
$EK_A^+$ = Encrypt with the public key of node A
$EK_B^+$ = Encrypt with the public key of node B
$EK_{sk}$ = Encrypt with the shared secret key
$EK_G$ = Encrypt with group key
$EK_{CH\text{-}BS}$ = Encrypt with the secret key shared between Cluster head and BS
R-type = Response type (positive or negative response)
R = Reason of negative acknowledgement
Intended-$ID_{dest}$ = ID of Intended Destination
Pk = public key
$ID_{src}$ = ID of source node
N = Nonce (Unique Random Number)
MAC = Message Authentication Code
CH = Cluster Head

The distribution of bits to different fields (as shown in table 2), introduces some upper limits, such as, size of source address is of 2 bytes, it means our LSec works only in the environment where number of sensor nodes not exceeding $2^{16}$. Length of Nonce (unique random number) field is of 3 bytes, so LSec can allow maximum of

$2^{24}$ connections at a time. The length of public key and private key is of exactly 128 bits and the length of secret key is of exactly 64 bits. Only stream cipher encryption algorithms are allowed to use because of a fixed length size of packets. MAC is of 64 bits.

**Table 2.** Distribution of bits to different fields of LSP

| Field | Size | Field | Size |
|---|---|---|---|
| Type | 4 bits | Public and Private key | 128 bits |
| IDsrc, IDdest | 16 bits | Secret key | 64 bits |
| Nonce (N) | 23 bits | token | 4 bytes |
| R-type | 1 bit | data | 30 bytes |

### 2.4 Procedure

LSec works in three phases, authentication and authorization phase, key distribution phase, and data transmission phase. Authentication and authorization is performed during the exchange of "Request" and "Response" packet by using symmetric scheme. Key distribution phase involves sharing of random secret key in a secure manner by using asymmetric scheme. In this phase "INIT" and "ACK" packets will be exchanged. Data transmission phase involves transmission of data packet in an encrypted manner.

Let's suppose node A wants to communicate with the node B. It will first send request packet to Base station, for receiving token and public key of node B. The request packet is encrypted with the secret key shared between node A and BS. BS first checks in the database via AzM that weather node A has rights to establish connection with node B. If yes, it generates the token which will be further used by the node B for the authentication of node A. That token is encrypted with secret key shared between node B and BS, so that node A will not able to decrypt token. BS will sent back a response packet that contains token, public key of node B and Nonce (Unique Random Number) that was there in request packet. Nonce will ensure node A that packet came from genuine BS. When node A gets the positive response from BS it sent the INIT packet to node B that contains Nonce, its own public key and token generated by BS. The whole INIT packet is encrypted with the public key of node B. When node B gets INIT packet it first check token, if it is correct, it will generate the secret key and sent it back to node A in an encrypted manner. When node A gets ACK packet, it deletes the public key of node B from its memory, and sent data to node B by using new session secret key. When data transmission complete, both nodes delete that session key. For group communication, each node uses the group secret key for data transmission in a secure manner. Cluster head will update this key after periodic interval.

## 3. Simulation and Performance Analysis

We have tested our LSec protocol on Sensor Network Simulator and Emulator (SENSE) [15]. In sensor node we introduce the middleware between application layer

and network layer as shown in figure 2.

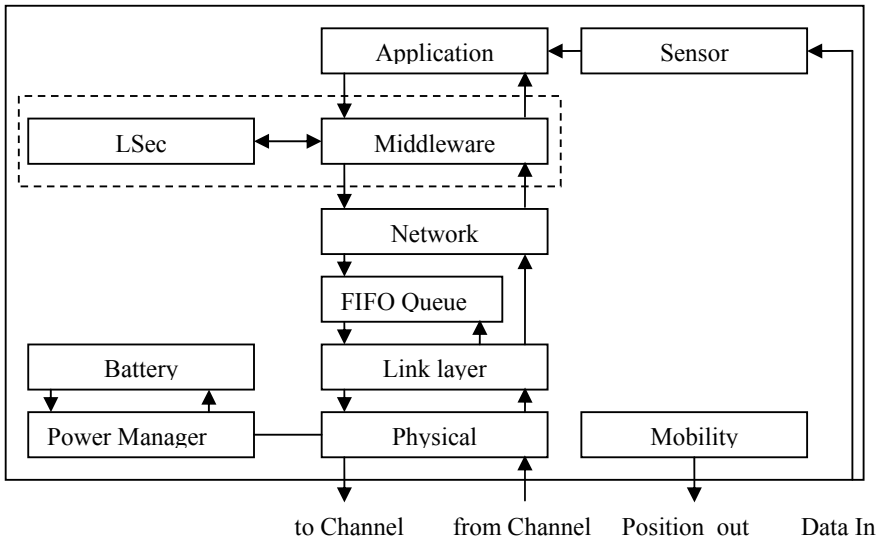


**Fig 2.** Sensor Node Architecture

That middleware uses LSec for the enforcement of security in the sensor network. At application layer we use constant bit rate component (CBR) that generate constant traffic during simulation between two communicating sensor nodes. For the demonstration and performance evaluation of LSec, CBR is run with and without LSec. We randomly deploy 100 sensor nodes plus one Base station (BS) in 1000 by 1000 terrain. Basic simulation parameters employed are described in table 3.

**Table 3.** Simulation Parameters

| | |
|---|---|
| Terrain | 1000x1000 |
| Total Number of Nodes | 101 (including BS) |
| Initial battery of each sensor node | $1x10^6$J |
| Power consumption for transmission | 1.6W |
| Power consumption for reception | 1.2 W |
| Idle power consumption | 1.15W |
| Carrier sense threshold | 3.652e-10W |
| Receive power threshold | 1.559e-11W |
| Frequency | 9.14e8 |
| Transmitting & Receiving antenna gain | 1.0 |

### 3.1 Performance Analysis of Communication Overhead

In our simulation scenario, application sent data packets of size 30 bytes in a periodic interval. The overall communication overhead of LSec for one to one communication

is decreases with the increase in transfer of number of data packets as shown in figure 3. Communication Overhead (C0 %) is calculated as

$$CO(\%) = \left(\frac{Nc * 74.125}{\sum_{i=1}^{n} N_i^P * 30}\right) * 100 \quad (1)$$

Where as 'Nc' is the total number of connections. $N_i^P$ is the number of packets transferred by node i. We multiplied 74.125 bytes to Nc because for every connection LSec exchange four control packets (Request, Response, Init, and Ack) during the authentication, authorization and key exchange phase whose cumulative size is 74.125 byte. Size of each data packet is 30 bytes.

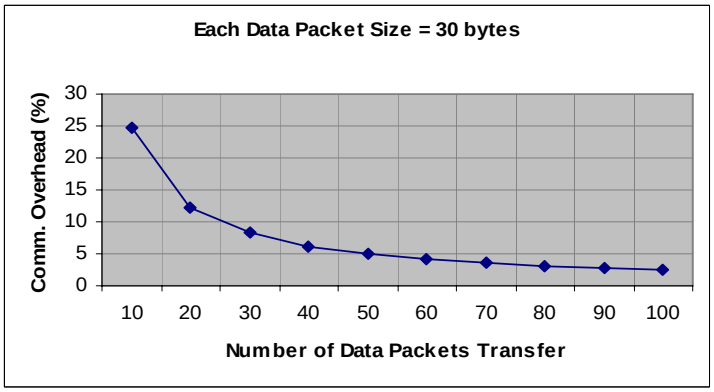


**Fig 3.** Communication Overhead (%) of LSec

### 3.2 Performance Analysis of Power Computation

Power Computation primarily depends upon the kind of symmetric and asymmetric scheme. If we assume that computation power required for symmetric encryption and decryption scheme is CSE and CSD respectively and computation power of asymmetric encryption and decryption scheme as CAE and CAD respectively. Then the total power consumption required by single node during first two phases is

$$Power\ Computation = (CSE + CSD) + (CAE + CAD) \quad (2)$$

Computation power required by a single node during data transmission phase is calculate as,

$$Power\ Computation = (TNSP*CSE) + (TNRP*CSD) \quad (3)$$

Where TNSP is the Total Number of Sent data packets and TNRP is the Total Number of received data packets.

### 3.3 Performance Analysis of Memory Consumption

Every sensor node needs to store only six keys, three of them are permanent and three are ephemerals. Permanent keys consist of one public key (self), one private keys and one public key of BS. Ephemerals keys consist of group key, public key of other node and session secret key. In order to save these keys only 72 bytes are needed. Details are given in table 4. This approach will make sensor network memory efficient.

**Table 4.** Storage Requirement of Keys

| S/No | Keys | Size (in bytes) |
|---|---|---|
| Permanent Keys | | |
| 1 | Public key of node | 16 |
| 2 | Private key of node | 16 |
| 3 | shared secret key b/w Node & BS | 8 |
| Ephemeral Keys | | |
| 4 | Group Key | 8 |
| 5 | Public key of other node | 16 |
| 6 | Session key | 8 |
| Total Storage size Required | | 72 bytes |

### 3.4 Performance Analysis of Energy Consumption

The main source of energy consumption at sensor node is its transmission and reception cost. We used SENSE that consumes energy in four different modes: TRANSMIT, RECIEVE, IDLE, and SLEEP. Energy consumption rate of each mode is given in table 3. For each connection, LSec exchange four control packets (Request, Response, Init, and Ack) of cumulative size 74.125 bytes that requires for authentication, authorization and key exchange mechanism. That is an acceptable tradeoff between energy and security. Simulation result of energy consumption is shown in figure 4.

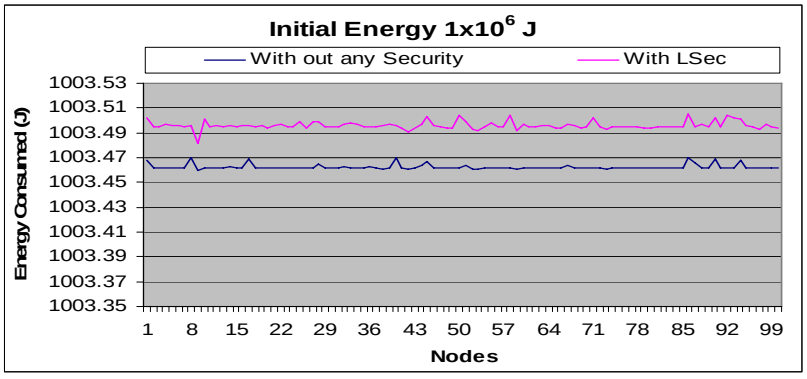


**Fig 4.** Energy Consumptions

### 3.5 Resilience against Node Compromise

Single node compromised will not expose the whole communication in network. Only the communication links that are established with compromised node will expose the network. Let's suppose 'Ncn' is the set of nodes that establish connections and 'Ncp' is the set of compromised nodes. Then Ncn ∩ Ncp will give us the set of nodes that are compromised as well as connected. Then the maximum number of connections that can be exposed only if all compromised nodes connected to uncompromised nodes. On the other hand minimum numbers of links that can be exposed only if all compromised nodes are connected with each other.

$$Max : Ncn \bigcap Ncp \tag{4}$$

$$Min : \begin{cases} \dfrac{Ncn \bigcap Ncp}{2} & for \longrightarrow even \\ \left(\dfrac{Ncn \bigcap Ncp + 1}{2}\right) & for \longrightarrow odd \end{cases} \tag{5}$$

If we assume that sensor networks consists of 1000 nodes and total 500 connections established between pair of nodes then the total links that can be minimum and maximum compromised is shown in figure 5.

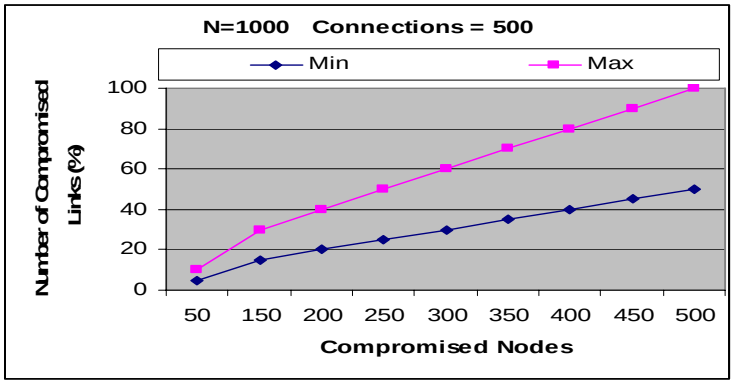


**Fig 5.** Percentage of Compromised Links

## 4. Comparison of LSec with Other Security Solutions

Comparison of all above discussed schemes with LSec is given in table 5. We provided comparison from the perspective of memory requirement, transmission cost, and some other basic security parameters such as authentication, authorization, confidentiality, etc. Data integrity is generally handled at link layer with the help of some hashing schemes such as MD5, SHA1 etc or by CRC schemes and availability is normally handled at physical layer. LSec lies between network and application

layer that's why it doesn't provide explicit data integrity and availability support.

**Table 5.** Comparison of LSec with other security solutions

| | | SPINS | TinySec | LiSP | LSec |
|---|---|---|---|---|---|
| Memory Requirement with respect to storage of keys | | 3 | Depended on KMS[1] | ≥ 8 | 6 |
| Transmission Cost | During key exchange (bytes) | -- | Depended on KMS | 12.6*TNN[2] | 74.125*TNC[3] |
| | During Data Transmission | 20% | 10% | > 20 | 8.33% |
| Public Key Cryptography Support | | No | No | No | Yes |
| Symmetric key cryptography Support | | Yes | Yes | Yes | Yes |
| Intrusion Detection mechanism | | No | No | Yes | Yes |
| Authentication support | | Yes | Yes | Yes | Yes |
| Authorization support | | No | No | Yes | Yes |
| Data Integrity support | | Yes | Yes | Yes | No |
| Confidentiality support | | Yes | Yes | Yes | Yes |
| Availability support | | No | No | Yes | No |

[1]KMS: Key Management Scheme
[2]KNN: Total Number of Nodes
[3]KNC: Total Number of Connections

## 5. Conclusion and Future Directions

We proposed Lightweight security protocol (LSec) for wireless sensor networks, which provides authentication and authorization of sensor node. It also provides simple secure key exchange scheme and confidentiality of data. LSec is highly scalable and memory efficient. It uses 6 keys, which takes only 72 bytes of memory storage. It introduces 74.125 bytes of transmission and reception cost per connection. It has the advantage of simple secure defense mechanism against compromised nodes. In future, we will try to solve the issue related to the neighboring nodes of the base station that suffered from higher communication overhead by forwarding request and response packets during authentication and authorization phase.

## References


1. C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", *proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications (WSNA'03)*, May 2003, pp. 113- 127
2. Wenliang Du, Jing Deng, Han, Y.S., Shigang Chen, Varshney P.K, "A key management scheme for wireless sensor networks using deployment knowledge", *proc. of INFOCOM 2004*, Mar 2004
3. Lydia Ray, "Active Security Mechanisms for Wireless Sensor Networks and Energy optimization for passive security Routing", *PhD Dissertation*, Dep. of Computer Science, Louisiana State University, Aug 2005
4. J. Kohl and B. Clifford Neuman, "The Kerberos Network Authentication Service (v5)", RFC 1510, Sep 1993
5. W. Diffie and M.E. Hellman, "New Directions in Cryptography", *IEEE Transaction on Information Theory*, vol. 22, Nov 1976, pp. 644-654.
6. R. L. Rivest, A. Shamir, L.M. Adleman, "A method for obtaining Digital Signatures and Public key cryptosystem", *Communication of ACM*, vol. 21(2), 1978, pp. 120-126
7. Erik-Oliver Blaß and Martina Zitterbart, "Towards Acceptable Public-Key Encryption in Sensor Networks", *proc. of 2nd International Workshop on Ubiquitous Computing, ACM SIGMIS*, May 2005
8. John Paul Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary, "Wireless sensor network security: A Survey", *Technical Report MIST-TR-2005-007*, July, 2005
9. A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. D. Tygar, "SPINS: Security protocols for sensor networks", *proc. of 7th annual international conference on Mobile computing and networking*, Rome, Italy, Aug 2001, pp 188-189
10. Chris Karlof, Naveen Sastry, and David Wagner, "TinySec: a link layer security architecture for wireless sensor networks", *Proc. of the 2nd international conference on Embedded networked sensor systems*, Baltimore, MD, USA, Nov 2004, pp 162-175
11. K. Jones, A.Wadaa, S. Oladu, L. W|son, and M. Etoweissy, "Towards a new paradigm for securing wireless sensor networks", *proc. of the 2003 workshop on New security paradigms*, Ascona, Switzerland, Aug 2003, pp 115 - 121
12. Taejoon Park, and Kang G. Shin, "LiSP: A Lightweight Security Protocol for Wireless Sensor Networks", *ACM Transactions on Embedded Computing Systems*, vol. 3(3), Aug 2004, pp. 634–660
13. Sencun Zhu, Sanjeev Setia, and Sushil Jajodia, "LEAP: Efficient Security Mechanism for Large-Scale Distributed Sensor Networks ", *Proc. of the 10th ACM conference on Computer and communications security*, Washington, USA, 2003, pp. 62-72
14. Elaine Shi and Adrian Perrig, "Designing Secure Sensor Networks", *IEEE Wireless Communications*, Dec 2004, pp. 38-43
15. Sensor Network Simulator and Emulator (SENSE) http://www.cs.rpi.edu/~cheng3/sense/