Implementation of LiSP using Park-Miller for Wireless Sensor Network

Kusumlata Jain
Department of Computer Science
Banasthali Vidyapith, Jaipur, India

Ankita Ojha
Department of Information Technology
Banasthali Vidyapith, Jaipur, India

ABSTRACT

Wireless sensor networks continuous to grow and a rosining technology in networks. WSN used in many critical situations and applications. One of major application of the sensor network is use in military. Due to these reason securities is a major challenge for WSN. Wireless sensor networks have many topics to do the research but security plays an important role in data and network security. In the present work, author use the block cipher rather than stream cipher, and key to perform encryption and decryption operation to create lightweight security protocol. Encryption algorithm takes two inputs, one is the text enter by the user another one is the dynamic key. Dynamic key has one most important advantage is that for each pair of the encryption and decryption new key is generated and previous one is discarded. In this paper dynamic key is generated by using park-miller random number generator. In the present work the key of 196 bits is used. This key is mainly used to convert the text entered by the user into the unreadable form. In this paper author divides the input plaintext into 49 bits of blocks. Dynamic key is following the concept of one time pad. Different key is used for the different parts of the plaintext so it is difficult and make it harder to break the security of data.

General Terms

Sensor Network, Park-miller

Keywords

Wireless sensor networks, dynamic key, park miller, encryption, decryption, one time pad, lightweight protocol.

1. INTRODUCTION

A sensor network consists of a large number of very small nodes that are deployed in some geographical area. The purpose of the network is to sense the environment and report what happen in the area it is deployed in. Sensor networks are used for surveillance and target tracking. A sensor network consists of various detection stations called sensor nodes, each of which is small and lightweight. All sensor nodes are equipped with a transducer, microcomputer, transceiver and power source. The transducer generates electrical signals based on sensed physical effects and phenomena. The microcomputer processes and stores the sensor output. The transceiver receives commands from a central computer and transmits data to that computer. The power for each sensor node is derived from a battery. Sensor node consist the limited computing power and memory space attacks are easily applicable over the WSN. To address these problem lightweight security protocol is created. [8]

This paper contained how to create lightweight security protocol by using park-miller random number generator algorithm. in this paper lightweight security protocol using park-miller random generator is proposed .In any security algorithm [1] it contains two inputs, one is the text entered by user and another one is the key. Key plays an important role in encryption. To generate the key pseudo random number is used. The Pseudo-Random numbers are plays important role in the wireless network or we can say that in communication system for protection of data from unauthorized access. The efficiency of the PRNG based cryptosystem is mainly based on the random key generated by its PRNG. [5]

Park-Miller algorithm is used to produce 32-bit sequences of key. The Park-Miller algorithm contains the following three criteria to produce good pseudo random numbers: sequence is satisfactorily Random, Sequence full period, Efficient Implementation with 32-bit arithmetic. [2]

The generated sequences of 32-bit sequence keys are acting an important role in the case of stream cipher to encrypt and decrypt plain text. In cryptography, a stream cipher [1] is asymmetric key [7] cipher where plaintext digits are combined with a pseudorandom cipher digit stream (key stream). In a stream cipher each plaintext digit is encrypted one at a time with the corresponding digit of the key stream, to give a digit of the cipher text stream. Finally the generated sequence of 32-bit pseudo random keys and by same sequence of keys used to encrypt and decrypt plain text of 196-bit simulation of proposed scheme is by using MATLAB. [2]

2. PROPOSED ALGORITHIM

In the following paper proposed a method based on the symmetric key encryption. In the proposed work every message is encrypted and decrypted by using the dynamic key than key is discarded after performing the encryption operation.

A key discarding function is called after every decryption function [4] and a new key is generated to perform the encryption and decryption. This concept is based on one time pad. The proposed algorithm is implemented on MATLAB 7.0.

2.1 Park – Miller Algorithm

Park-Miller algorithm [2] is based on the LCG:

$X_{n+1} = A X_n \bmod (2^{31}-1)$.

Where [1] the constant value “A” chosen as 15,887. Constant

A=158557; $(2 < A < M-1)$

M=2157483747; $(2^{31}-1)$

q=13607; (M div A)

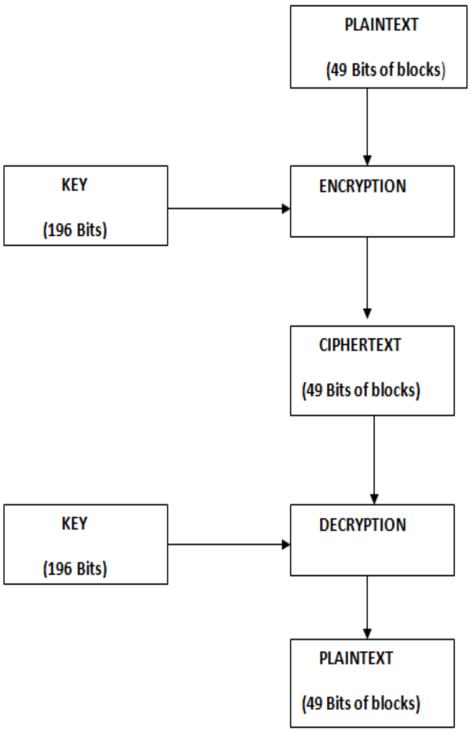
r=157205; (M mod A)

Var

High: =in_seed div q;

Low: =in_seed mod q;

```
test: =a*low –r*high;
if test > 0 then
Out_seed:= test;
random: =out_seed / m;
else
Out_seed:= test +m;
random: =out_seed / m;
end;
```



. **Fig 1: Planned Method**

2.2 Dynamic Key

In the planned work like the one time pad or we can say that dynamic key is generated by using linear park-miller random number generator. In planned work user enter a key. The size of the input key is 32-bit. An inbuilt key 'ibk' is concatenated with the key which is entered by user to creating a matrix of 14X14. A random function (FR$_r$) is used to produce the key. A random function contains various matrix operations like addition, multiplication, shifting etc. In which the random number X is added to the final matrix to produce the dynamic key. [6][7]

2.2.1 Key Generation Algorithm

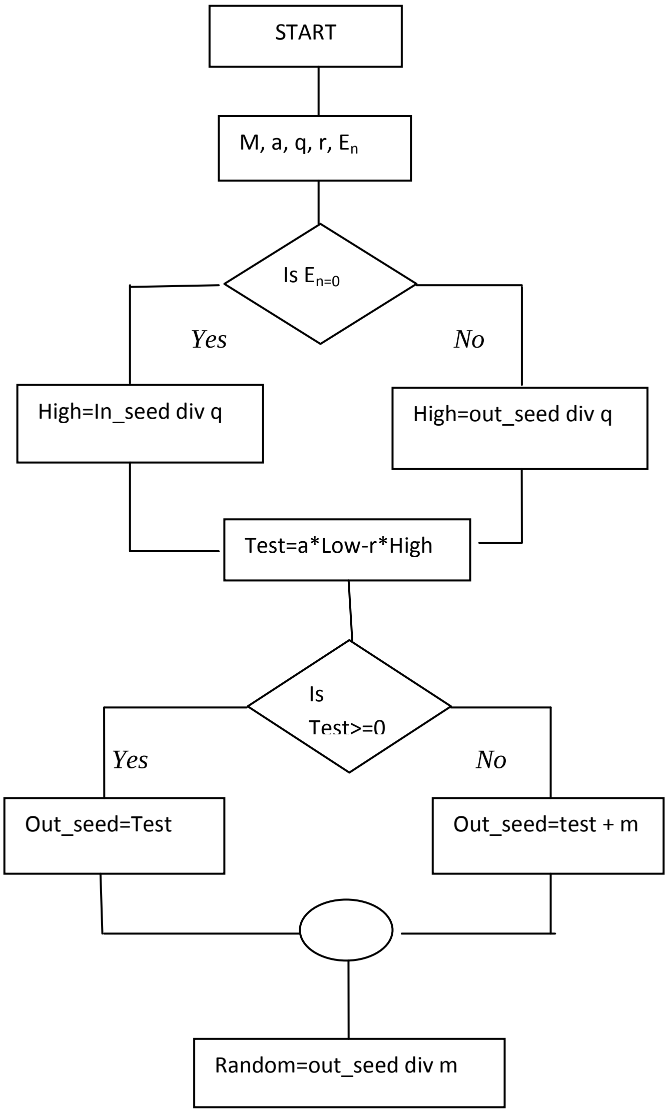


Fig. 2 Flow chart of Park-miller algorithm

Steps of the algorithm are as follows:

Initialize the input in_seed and parameters

1. a = 158577, m= 2157483747, q = 13607, r = 157205.
2. calculate the value of High = in_seed div q and Low = in_seed mod q.
3. Then calculate the equivalent test value.
4. If test > 0, save test as new out_seed, otherwise save test + m.
5. Output the new out_seed.
6. Iterate, and let the out_seed be the new in_seed.

2.2.2 MATLAB Implementation for Randomize Function (FR$_r$)

This randomize function use the different array to calculate the the dynamic key DSK.

```
P  [sl] [1] = EK;
% Text entered by user.
Q [1] [sl] = EK;
%Text entered by user.
R [14] [1] = find(ibk,14,'first') ;
% find first 14 character of ibk.
S [1] [14] = find(ibk,14,'last') ;
% last 14 character of ibk.
T [14] [sl] = P [14] [1] * Q [1] [sl];
U [sl] [14] = A [sl] [1] *S [14] [1] ;
V [14] [14] = T [14] sl[] * U [sl] [14] ;
W [14] [14] = V [14] [14] + $X_1$ ;
DSK  = W [14] [14] ;
%  DSK is a dynamic secret key.
```

2.2.3 Determine the number of plaintext blocks

```
PT = Text enter by users
S = length (PT)
% Length of the plaintext enter by user.
KI = S mod 49 ;
I = 49 –KI ;
  % number of padding bits
NPT = (PT - KI) / 49;
% number of plaintext blocks of 49 bits.
If (KI==0)
NP = NPT
Else
NP = NPT + 1
For (j = 1 to N)
{
CALL ENCRYPTION PROCESS
}
```

3. ENCRYPTION PROCESS

This paper proposes an encryption process which contains four rounds. In each round different key is applied to different parts of the plain text. The encryption process contains many operations likes Addition, XOR, Transpose, operation. The block size of the text enter by user for the encryption is 49 bits. The size of the dynamic key which is used to convert the plaintext into ciphertext is 196 bits.[4]

Initially the text enters for encryption is divided into 49 bits of blocks. Dynamic key size which is used for encryption and decryption is 196 bits which is the 14X14 matrix. The dynamic key for first round is divided into four parts: DP1, DP2, DP3, and DP4. [4]

DP1 the first part of the key is applied to the first part of the text which is entered by user to perform the encryption operation and at the end of the round 1 second part of the key DP2 is applied over the last part of the plaintext. The operations which are performed over the plaintext is ADD, XOR and TRANSPOSE. The output produces by round 1 is called round1 ciphertext (RP1).

In the next round of encryption process dynamic key is divided into two parts: DKF and DKL. DKF contains the first 64 bits of the key and DKL contains the last 147 bits of the key. Futher DKL is divided into two parts DKL1 and DKL2. Further DKF is divided into 3 parts these are DKF1, DKF2, DKF3.

In the second round only one key is used to perform the encryption and decryption operation. The output of the second round is known as round 2 ciphertext (RP2).

In the next round or we can say that in third round two keys are used one is DKL2 and another is DKF1 is applied over the output of the round 2.This round contain the same operation like Addition, subtraction , XOR, Transpose.

In last or we can say that in fourth round also contains the two keys one is DKF2, and another one is DKF3 which are applied over the output of the second round. The output produces by last round is known as round 4 ciphertext.

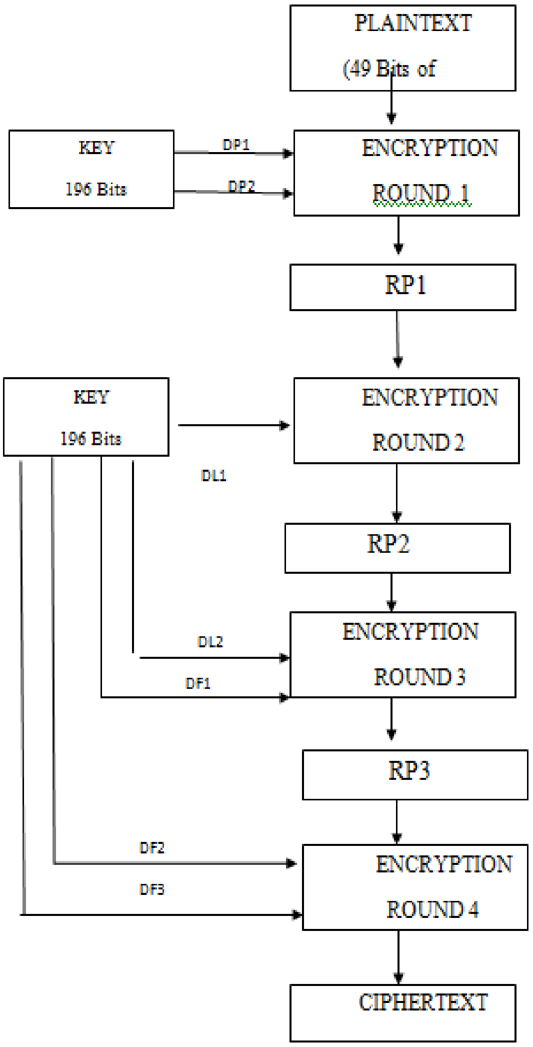


Fig. 1 Encryption process for lightweight protocol

4. DECRYPTION PROCESS

The decryption method is inverse of the encryption method. This process converts the unreadable message into the original message. As the encryption process decryption method contains the four rounds to perform the decryption method. Decryption method contains the same operations which are used in the encryption process like subtractions, XOR, Transpose. In the first round of the decipherment process operations are applied over the ciphertext which produce the first round ciphertext which is denoted by RCT1. In the next round of the decipherment process operations are applied over the ciphertext which produce the second round ciphertext which is denoted by RCT2. . In the third round of the decipherment process operations are applied over the ciphertext which produce the second round ciphertext which is denoted by RCT3. . In the last round of the decipherment process or we can say that at fourth round, operations are applied over the ciphertext which produce the fourth round output which is the plaintext or original message.[4]

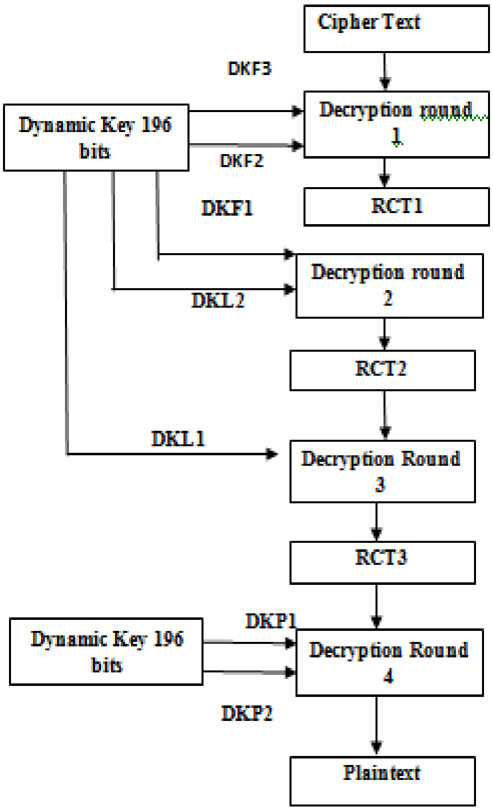


Fig. 4 Decryption process for lightweight protocol

5. COMPARE THE PARFORMANCE AND TIME COMPLEXITY

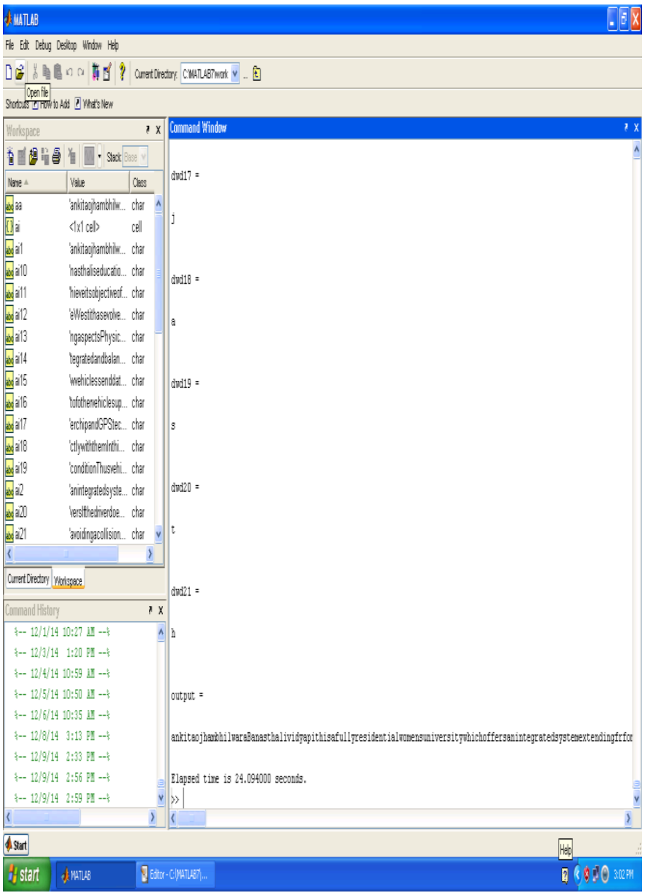
This section describes the key size, block size of different algorithm which is used to perform encryption and decryption. It also describe the time taken by various algorithm to perform encryption and decryption. In this paper lightweight protocol using LCG and lightweight protocol using park miller both implemented in MATLAB 7.0 and specification of computer in which algorithm is run is Intel Core 2 Duo CPU, 2.96 GB of RAM. When we run protocols on this system than time taken by both the lightweight protocols have time difference so we can say that LiSP using park-miller is more efficient than LiSP using LCG.

Table 1: Key and blocksize of algorithim

Algorithm	Block Size(Bits)	Key Size (Bits)
DES	64	64
LiSP using LCG	196	49
Proposed Algorithm	196	49

Table 2:Time complexity of different algorithim

Algorithm	Kilobytes processed	KB/Second
LiSP using LCG	12	24.094
Proposed Algorithm (LiSP using Park-Miller algorithm)	12	20.250

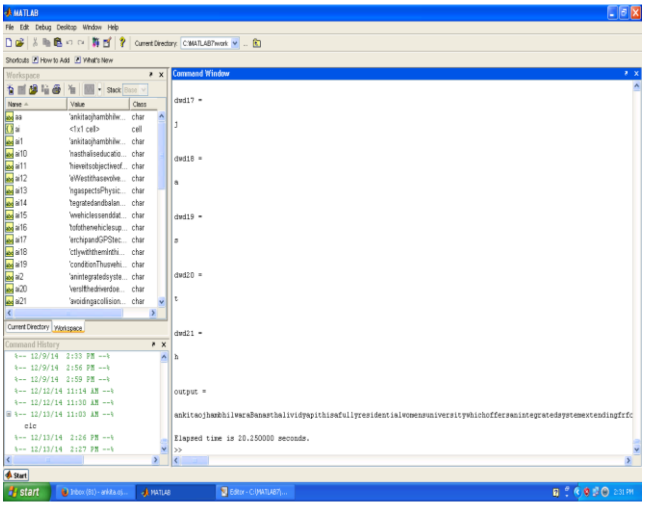

MATLAB
File Edit Debug Desktop Window Help
Current Directory: C:\MATLAB7\work
Open file
Shortcuts How to Add What's New
Workspace
Stack: Base
Current Directory Workspace
Command History
Command Window
Start
start MATLAB Editor - C:\MATLAB7\...
Help
3:02 PM

Name ▲	Value	Class
aa	'ankitaojhambhilw...	char
ai	<1x1 cell>	cell
ai1	'ankitaojhambhilw...	char
ai10	'nasthaliseducatio...	char
ai11	'hieveitsobjectiveof...	char
ai12	'eWestithasevolve...	char
ai13	'ngaspectsPhysic...	char
ai14	'tegratedandbalan...	char
ai15	'wehiclessenddat...	char
ai16	'tofthevehiclesup...	char
ai17	'erchipandGPStec...	char
ai18	'ctlywiththemInthi...	char
ai19	'conditionThusvehi...	char
ai2	'anintegratedsyste...	char
ai20	'versIfthedriverdoe...	char
ai21	'avoidingacollision...	char

```
%-- 12/1/14 10:27 AM --%
%-- 12/3/14  1:20 PM --%
%-- 12/4/14 10:59 AM --%
%-- 12/5/14 10:50 AM --%
%-- 12/6/14 10:35 AM --%
%-- 12/8/14  3:13 PM --%
%-- 12/9/14  2:33 PM --%
%-- 12/9/14  2:56 PM --%
%-- 12/9/14  2:59 PM --%
```

```
dwd17 =

j


dwd18 =

a


dwd19 =

s


dwd20 =

t


dwd21 =

h


output =

ankitaojhambhilwaraBanasthalividyapithisafullyresidentialwomensuniversitywhichoffersanintegratedsystemextendingfrfor

Elapsed time is 24.094000 seconds.
>>
```

Fig 5. Result of LiSP using LCG


MATLAB
File Edit Debug Desktop Window Help
Current Directory: C:\MATLAB7\work
Shortcuts How to Add What's New
Workspace
Stack: Base
Current Directory Workspace
Command History
Command Window
Start
start
Inbox (81) - ankita.oj...
MATLAB
Editor - C:\MATLAB7\...
2:31 PM

Name	Value	Class
aa	'ankitaojhambhilw...	char
ai	<1x1 cell>	cell
ai1	'ankitaojhambhilw...	char
ai10	'nasthaliseducatio...	char
ai11	'hieveitsobjectiveof...	char
ai12	'eWestithasevolve...	char
ai13	'ngaspectsPhysic...	char
ai14	'tegratedandbalan...	char
ai15	'wehiclessenddat...	char
ai16	'tofthevehiclesup...	char
ai17	'erchipandGPStec...	char
ai18	'ctlywiththemInthi...	char
ai19	'conditionThusvehi...	char
ai2	'anintegratedsyste...	char
ai20	'versIfthedriverdoe...	char
ai21	'avoidingacollision...	char

```
%-- 12/9/14  2:33 PM --%
%-- 12/9/14  2:56 PM --%
%-- 12/9/14  2:59 PM --%
%-- 12/12/14 11:14 AM --%
%-- 12/12/14 11:30 AM --%
%-- 12/13/14 11:03 AM --%
  clc
%-- 12/13/14  2:26 PM --%
%-- 12/13/14  2:27 PM --%
```

```
dwd17 =

j


dwd18 =

a


dwd19 =

s


dwd20 =

t


dwd21 =

h


output =

ankitaojhambhilwaraBanasthalividyapithisafullyresidentialwomensuniversitywhichoffersanintegratedsystemextendingfrfc

Elapsed time is 20.250000 seconds.
>>
```

Figure 6. Result of (LiSP using Park-Miller algorithm

6. CONCLUSION

In this paper lightweight protocol is created using dynamic key for wireless sensor network. In this dynamic key is generated by using park miller random number generator algorithm. Park miller algorithm generates pseudo random number. The main strong point of this method is dynamic key which generate every time after encryption and decryption process. Par miller algorithm provides the best authentication mechanism. In the proposed method the key size is 49 bits and block size is 196 bits which is similar to lightweight protocol using LCG, but proposed is completed in less time than LiSP LCG. So proposed method is more secure than existing. This type of lightweight protocol can meet our security requirement.

7. REFERENCE

[1] William Stalling, "Applied Cryptography" 4th ed.

[2] Bharatesh N, Rohith S, "FPGA Implementation of Park-Miller Algorithm to Generate Sequence of 32-Bit Pseudo Random Key for Encryption and Decryption of plain text", International Journal of Reconfigurable and Embedded Systems (IJRES), Vol. 2, No. 3, November 2013, pp. 99~105.

[3] Bo Sun, Chung-Chih Li, Kui Wu, Yang Xiao, "A lightweight secure protocol for wireless sensor networks", ELSEVIER, computer communication, 29 (2006) 2556–2568.

[4] Zeenat mahmood, Anurag jain, Chetan agrawal," Hybridize Dynamic Symmetric Key Cryptography using LCG", In International Journal of Computer Applications (0975 – 8887) , vol. 60-no. 17 ,December 2012 .

[5] SK Park, KW Miller, Random number generators: Good ones are hard to find. Communications of the, *ACM*. 32(10): 1192-1201.

[6] Che-Chens Lin, Shiuhyng Shieh, Jia-Chun Lin, "Distributed Key Agreement Protocol for Wireless Sensor Network", The Second International Conference on Secure System Integration and Reliability Improvement.

[7] Suman Bala, Gaurav Sharma and Anil K. Verma, "Classification of Symmetric Key Management Schemes for Wireless Sensor Network", International Journal of Security and Its Applications Vol. 7, No. 2, March, 2013.

[8] Priya L C, Shantal Devi Patil, " A survey on Sensor Aunthication in Dynamic Wireless Sensor Networks ", International Journal of Computer Science and Information Technology Research,Vol. 2, Issue 2, pp: (454-461), Month: April-June 2014.

[9] Delan Alsoufi, Khaled Elleithy, Tariq Abuzaghleh and Ahmad Nassar " Security in wireless Sensor Network", International journal of computer Science and Engineering Survey, vol 3, no. 3, June 2012.

[10] Anupma Sangwan1, Deepti Sindhu2, Kulbir Singh, " A Review of various security protocols in wireless sensor networks", Anupma Sangwan et al, int. j. comp. tech. Appl., Vol 2 (4), 790-797.