A Symmetric Key Block Cipher to Provide Confidentiality in Wireless Sensor Networks

BANNISHIKHA BANERJEE[1]
JALPA T. PATEL[2]

Department of Computer Engineering, Gujarat Technological University, India
[1]bannibanerjee@gmail.com, [2]pateljalpa_t@yahoo.com

Abstract. The sensors of wireless sensor networks communicate by sending and receiving packets among one another. It is important to provide security in wireless sensor networks so that only the valid or legitimate user gets the message. Cryptography is an important concept which provides security in wireless sensor networks. It encrypts the data and provides diffusion and confusion to hide the ciphertext's relationship with the plaintext and the key. In this paper, we have proposed a symmetric key block cipher to provide confidentiality in wireless sensor networks. We simulated sensor network in network simulator (ns – 2.35) using AODV protocol. Our proposed approach is implemented in AODV protocol and the performance is compared with other existing approach like genetic operations. Comparison is done on the basis of security, bit flip rates and occurrence avalanche effect. We noticed that, our proposed approach provides around 50 – 60% bit flip rate which is much higher than genetic operations, which is around 12.5%.

Keywords. Wireless sensor networks, cryptography, plaintext, ciphertext, diffusion, confusion, avalanche effect.

(Received February 27, 2016 / Accepted June 9, 2016)

1 Introduction

Wireless Sensor Networks are heterogeneous systems that contain many small nodes called sensors. These are usually used to monitor surrounding environment and have large applications which are sensitive in nature. These applications include monitoring, tracking and controlling areas like object or human activities. These include battlefield monitoring, forest monitoring, road traffic monitoring, temperature sensing etc. Many of the applications collect and maintain the data. Wireless Sensor Networks communicate by sending and receiving packets among one another. It is important to provide security in wireless sensor networks so that only the correct user gets the message. Cryptography is an important concept which provides security in Wireless Sensor Networks.

The major parameters for providing confidentiality are [2]:

Authentication: It is the process of confirming that the user or the node is who it claims to be. This is done by using face recognition, finger prints, passwords, security questions, digital signatures etc.

Confidentiality: It is the process of making sure that no one, other than the authorized user can read the message. This is done by changing the plaintext by encrypting it.

Integrity: It is the process of making sure that the data received by the receiver is exactly what it was sent by the sender. It did not get modified or tampered with. This is done by using hash functions.

Data Freshness: It is the process of making sure that the data received is the recent data and not some stray or old one. This is done by providing timestamps.

In this paper, we have proposed a symmetric key block cipher to provide confidentiality in wireless sensor networks.

Confidentiality is the process of making sure that the message can be read only by the legitimate or the valid user and not any other user or attacker. The secrecy of the message can be maintained by using cryptography. The message or plaintext is encrypted, using a key and the cryptographic algorithm.

This generates a ciphertext, which is transferred from the source node to the destination node. Since the packet is encrypted, so even if a malicious user gets its hands on the packet, it still cannot understand it, or decrypt it without the key. The packet is received at

the destination and then it is decrypted using the same key. This is called symmetric encryption. In asymmetric encryption, the keys used for encryption and decryption are different. Our proposed approach used the same key and so it is symmetric key cryptographic approach.

The key parameter in measuring security provided by an algorithm is diffusion and confusion. Diffusion is the property, in which, when one bit of the plaintext changes, several bits of the ciphertext should get changed.
If the number of bits changing in the ciphertext is at least 50% of the total length of the ciphertext, then we can say that avalanche effect has occurred. Confusion is the property that hides the relationship between key and ciphertext in the similar manner. In the proposed scheme, we chose several plaintexts – key pairs and compared the ciphertext, by changing single bits in plaintext or key. Then we compared the bit flip rate, with existing approaches. In section 2 of this paper, we surveyed and studied various other encryption algorithms. Among those algorithms, we chose to compare our results with genetic approach [4] and RC5 [9] on the basis of avalanche effect and time taken to encrypt and decrypt. Section 3 explains our proposed scheme. Section 4 and 5 shows results and implementation details of our proposed scheme. In Section 6 we compare it with genetic approach. Finally, in Section 7 we conclude our work.

2 Related Works

In [3], they proposed a lightweight encryption scheme for tiny sensor devices that guarantees data confidentiality between the source and the destination nodes. They presented a simple pseudorandom bit sequence generation scheme using elliptic curve points that does not involve any floating point calculation. Therefore, it avoids the problem of precision loss and also minimizes the computational costs for low power sensor nodes. Then they proposed a cryptosystem that generates a different pseudorandom bit sequence for every new session and preserves independent behavioral characteristic of the algorithm. Then, the proposed scheme is compared to RC5 and Skipjack in terms of memory occupation, operation time and energy efficiency.

But their proposed scheme does not provide sufficient, diffusion and confusion or avalanche effect. Although it is lightweight, compared to other ciphers, but the bit flip rates is lower than others.

In [4], the mutation and crossover genetic operations are used as tools for introducing diffusion and confusion properties in the ciphertext. The mutation process is applied to create random diversity (diffusion) in the ciphertext, whereas the crossover operation is used to change the order of the mutated text or image data (confusion). The main benefit of using genetic operations is that they provide relatively sufficient diversity in the ciphertext.

But in this approach as well, the amount of security is not sufficient, although it is energy efficient. Avalanche effect does not occur in this approach; hence it is not very secure.

In [5], the author proposed a unique, dynamic cryptographic algorithm to provide security to the wireless sensor network by securing each and every individual nodes of the network. Their proposed cryptographic algorithm provides nodal level security in a wireless sensor networks. Hence it makes the vital data and information stored in each of the nodes secure enough to allow the use of simpler and more energy efficient routing protocols for data transmission and communication among nodes, over the network.

All the nodes are embedded with a single network wide shared key, say *K1*. *K1* is used to transmit the message using a symmetric encryption protocol such as HIGHT or PRESENT or any other lightweight cryptographic method designed for wireless sensor networks. All the nodes contain the sensed information from the environment that they are monitoring. Since the sensor node does not process the sensed data itself, so they encrypted this data as soon as it is gathered from the environment using the key *K1* and the cryptographic algorithm used in message transmission.

In [6], they proposed a security approach which uses Secret Key *(SK)* cryptography and key management along with re – keying support. A salient feature of this approach is that a *SK* is embedded in the source code of every node to protect the other keys in its non – volatile memory. Even if the node gets captured physically by a malicious user; the sensitive information cannot be retrieved. Their key selection protocol uses the node ID and some basic rotate and multiplication function to select the key for current data transmission. Simulation results showed that this security mechanism efficiently controls the various attack with lower resource requirements and the network resilience against node capture is substantially improved.

In [7], they proposed a light weight energy efficient

encryption algorithm (LEE) for tiny embedded devices, such as sensor network nodes. They presented experimental results of LEE under real sensor nodes operating in TinyOS. They also discussed the secrecy and privacy of their algorithm by presenting a security analysis of various tests and cryptanalytic attacks. This encryption algorithm does not use complicated operators or any s – Boxes. But although this scheme is light weight and uses low energy, it still does not provide enough diffusion and confusion to ensure security.

In [9], they discussed DES and Blowfish encryption algorithms. There are many different types of encryption methods, for producing cipher text like stream cipher (such as RC5) and block cipher (such as DES, blowfish and so on).

The above ciphers are currently being used. We compared our proposed scheme with genetic operations.

3 Proposed Scheme

First of all, a 256 bits key is generated and it is divided in two subkeys, left half and right half. We have a plaintext of block size 128 bits. This plaintext is XORed with the most significant, i.e. left half of the key. Now the XORed plaintext is divided into individual bytes.

Now we choose the most significant byte of the right half of the key and count the number of 1s. If the number of 1s in the byte is even, then swap each bit of the corresponding plaintext byte with its consecutive bit. If the number of 1s in the byte is odd, then swap each bit of the corresponding plaintext byte with the bit next to its consecutive bit.

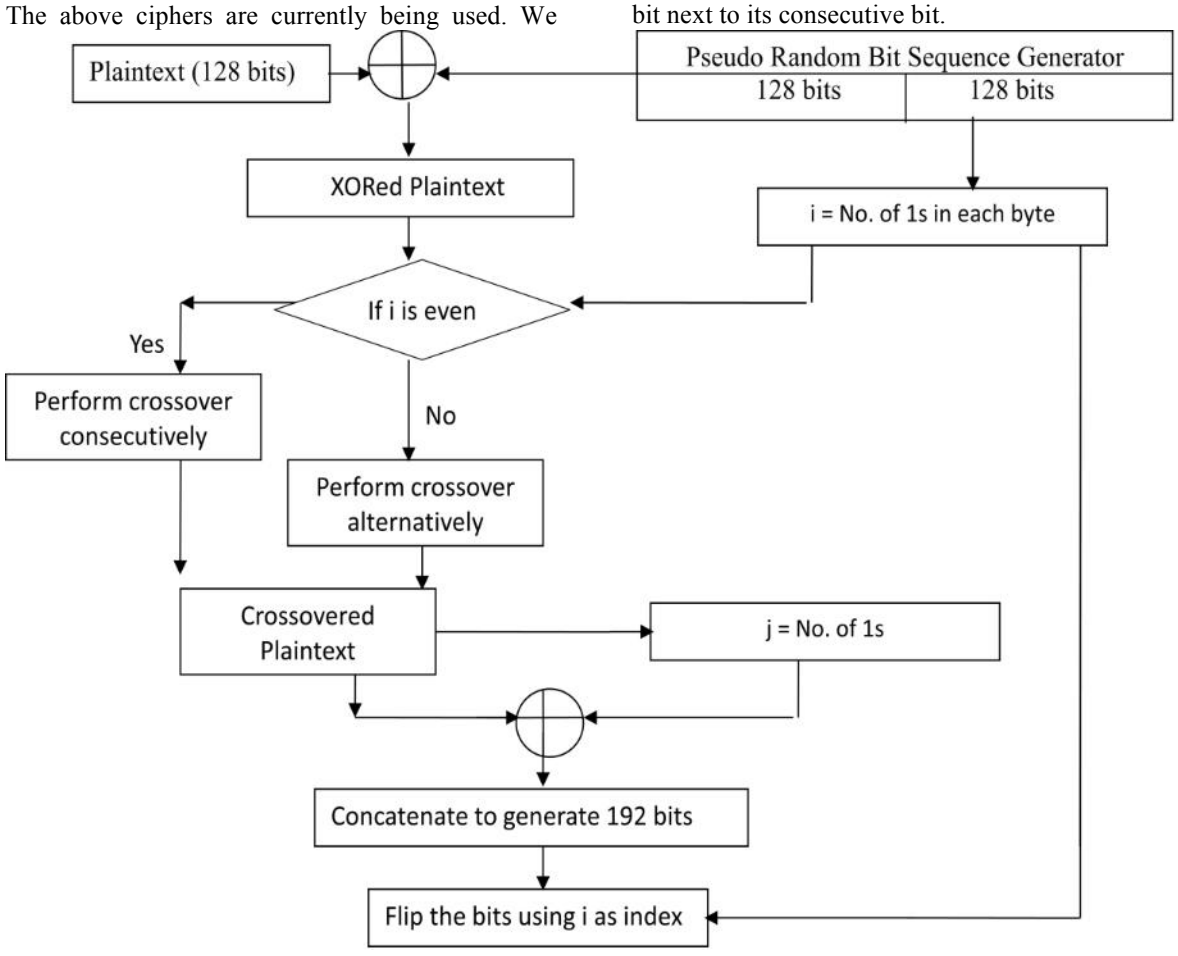

Figure 1: Proposed Scheme

Algorithm 1: XOR Operation and Crossover

1: Divide the 128 bits plaintext and 256 bits key into 8 bits and 16 bits block size, P[m] and K[n], respectively.
2: for i = 0 to m do
X[i] = P[i] xor K[i] 3: for i = m to m+8
if K[i] = 1 then
count1++
4: if count1 is even then
for i = 0 to m
swap(p[i],p[i+1]); // skip the already swapped bits
else if i is odd then for
i = 0 to m
swap(p[i],p[i+2]); // skip the already swapped bits
5: Store the crossover plaintext in cross[m]

Now count the number of bits of the crossover plaintext. Convert this number into 4 bits binary and concatenate with itself to get an 8 bits count array. For instance, suppose the number of 1s in crossover plaintext is 5 then corresponding binary is 0101. Now concatenate 0101 with itself to get 01010101, hence an 8 bits count array. Perform XOR operation on the crossover plaintext with the count array. Then concatenate the four bit count (0101) to the XORed result.

Algorithm 2: XOR operation

1: for i = 0 to m
if cross[i] = 1 then
count2++
2: for i = 0 to m
cross[i] = cross[i] xor count[i]
3: Concatenate(cross, count) ⑦ mut

Now count the number of 1s in the corresponding byte of the right half of the key. Then use this as the index to complement the bits. For instance, if the number of 1s in the corresponding byte of the right half of the key is 5, then flip the bits of the 12 bits result from that index. The resulting 12 bits is the ciphertext. Now do this for the entire 128 bits block of the plaintext to generate a 192 bits ciphertext.

Algorithm 3: Complement operation or mutation

1: for i = m to m+8
if K[i] = 1 then
count1++
2: for i = count to m+4
if mut[i] = 0 then
mut[i] = 1
else if mut[i] = 1 then
mut[i] = 0
3: mut[0]…mut[12] is the ciphertext.

The decryption process is the exact reverse of the encryption process, which is done using the same key. Hence the proposed scheme is symmetric key encryption.

4 Results

Plaintext Sensitivity Test: Plaintext sensitivity test is the way of ensuring diffusion provided by a given algorithm. Assume that we have plaintext, key pairs and we encrypt it using the proposed approach, to generate ciphertext. Now if we change 1 bit of the plaintext by keeping the key fixed, a significant number of bits in the ciphertext must change, to ensure diffusion. In our proposed scheme, when we changed one bit of the plaintext, we noticed significant amount of change in ciphertext. At least 50% bits are changing, during this test. This is more than genetic approach, as seen from the following tables.

Table 1: Plaintext Sensitivity Test (Genetic [4])

Plaintext 1	Plaintext 2	Key	Bit Flip
00011001	10011001	1101010110100011	12.50%
00000000	10000000	1001110001011011	12.50%
11111111	11110111	1110001001100000	12.50%
11110000	11010000	1011010100011010	12.50%
10101010	10111010	1100001001000001	12.50%

Table 2: Plaintext Sensitivity Test (Proposed)

Plaintext 1	Plaintext 2	Key	Bit Flip
00011001	10011001	1101010110100011	66.66%
00000000	10000000	1001110001011011	58.33%
11111111	11110111	1110001001100000	58.33%
11110000	11010000	1011010100011010	58.33%
10101010	10111010	1100001001000001	66.66%

Key Sensitivity Test: Key Sensitivity test is the way of ensuring confusion provided by a given algorithm. Assume that we have plaintext, key pairs and we encrypt it using the proposed approach, to generate ciphertext. Now if we

change 1 bit of the key by keeping the plaintext fixed, a significant number of bits in the ciphertext must change, to ensure confusion. In our proposed scheme, when we changed one bit of the key, we noticed significant amount of change in ciphertext. More than 50% bits are changing, during this test. In Table 3, the corresponding plaintext from Table 1 is used, for different keys.

Table 3: Key Sensitivity Test (Genetic [4])

Key 1	Key 2	Bit Flip
1001010110100011	1101010110100011	12.50%
1001100001011011	1001110001011011	12.50%
1111001001100000	1110001001100000	12.50%
1011011100011010	1011010100011010	12.50%
1101001001000001	1100001001000001	12.50%

Table 4: Key Sensitivity Test (Proposed)

Key 1	Key 2	Bit Flip
1001010110100011	1101010110100011	66.66%
1001100001011011	1001110001011011	66.66%
1111001001100000	1110001001100000	58.33%
1011011100011010	1011010100011010	58.33%
1101001001000001	1100001001000001	66.66%

The above tables, Table 1and 2 show that when we change 1 bit of the plaintext, the proposed scheme shows highest change in the ciphertext. This shows that the proposed scheme shows highest avalanche effect and therefore is secure. Genetic operations do not show avalanche effect.

The tables, Table 3 and 4, show that when we change 1 bit of the key, the proposed scheme shows highest change in the ciphertext. This shows that the proposed scheme shows highest avalanche effect and therefore is secure. Genetic operations do not show avalanche effect on the basis of key sensitivity because it has a lower key sensitivity percentage (<50%), as compared to the proposed scheme.

To make sure that the bit flip rate is at least 50%. We selected several plaintext key pairs and generated corresponding ciphertext. Then we flipped 1 bit of the plaintext or key to see the change in ciphertext. We noticed that, more than 50% (see Table 1) of the bits of the ciphertext is changing, when just 1 bit of the plaintext or key is changed. This confirms the occurrence of avalanche effect.

The proposed scheme shows completeness which means that the each and every bit of ciphertext depends on all the bits of input. This means that, if we change just 1 bit of the input, i.e plaintext or key, then at least 50% of bits of the ciphertext should get changed. An encryption algorithm is said to be complete if it satisfies, strict avalanche effect. Therefore, occurrence of avalanche effect, assures that our proposed approach satisfies the completeness property.

5 Implementation Details and Outputs

Implementation of this work is carried out using programming language O – TCL in AODV protocol. The tool used is Network Simulator ns – 2.35 in Linux based operating system Ubuntu. Plaintext of size 8 bits is send from the source node, to the destination node. A plaintext is chosen, and it is encrypted using genetic approach and the proposed scheme. Following figures show the output screenshots of encryption and decryption process using genetic approach and the proposed approach.

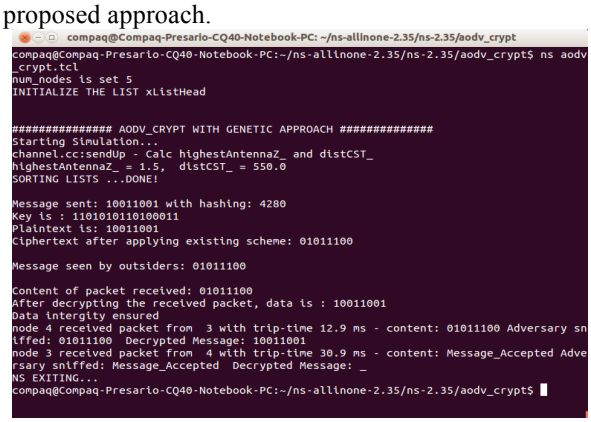

Figure 2: Encryption 1 using genetic approach

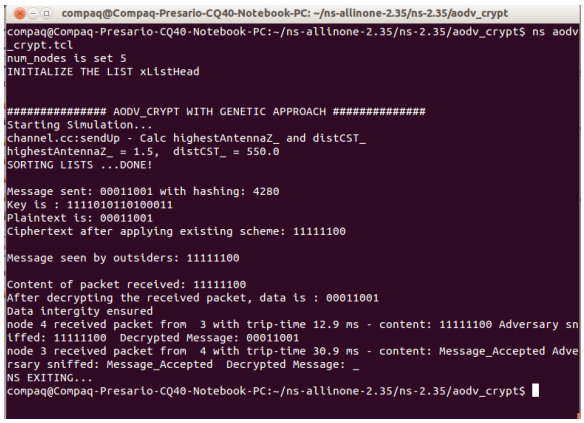

Figure 3: Encryption 2 using genetic approach.

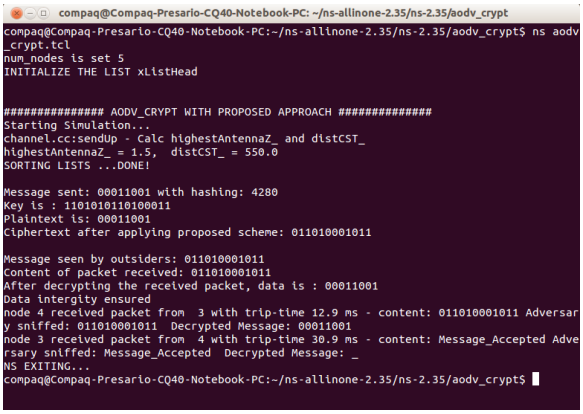

Figure 4: Encryption 1 using proposed scheme.

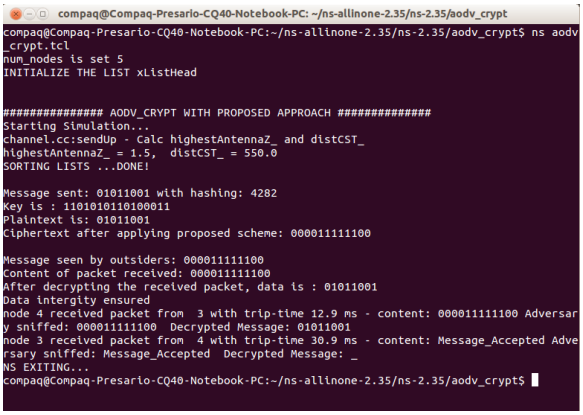

Figure 5: Encryption 2 using proposed scheme.

From the above figures (see Figure 2 and Figure 3 for genetic approach and see Figure 4 and Figure 5 for proposed scheme) we can see that in proposed scheme if one bit of the plaintext is changed then, more than 50% bits of the ciphertext is changing. Therefore providing strong avalanche effect. Hence we can say that our proposed algorithm is secure, when it comes to providing data confidentiality. Our proposed scheme shows higher bit flip rate as compared to genetic approach, hence higher avalanche effect.

6 Comparative Analyses

The following figures show the bit flip percentage, based on plaintext and key. Five files are chosen then their plaintext sensitivity percentage and key sensitivity percentage are calculated. Then Xgraphs are plotted to represent the same.

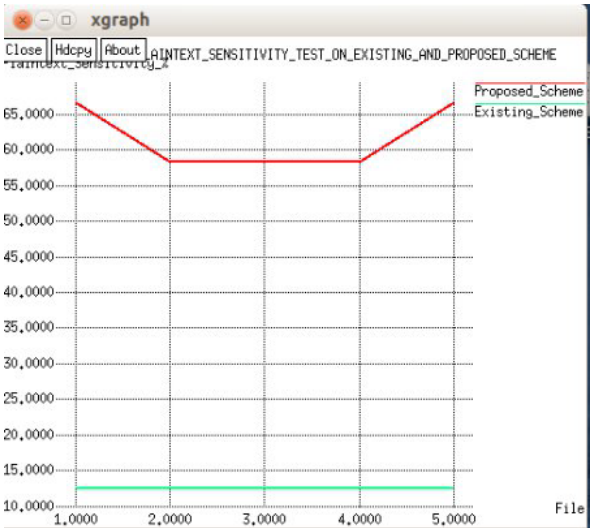

Figure 6: Plaintext Sensitivity Percentage %

Plaintext sensitivity test and key sensitivity test are the ways of ensuring diffusion and confusion provided by an algorithm. Assume that we have plaintext, key pairs and we encrypt it using the proposed approach, to generate ciphertext. Now if we change 1 bit of the plaintext or key by keeping the other constant then at least 50% of the bits should change. From Figure 8 and Figure 9 we can see that the proposed scheme shows higher plaintext and key sensitivity as compared to genetic approach and RC5 hence it is secure.

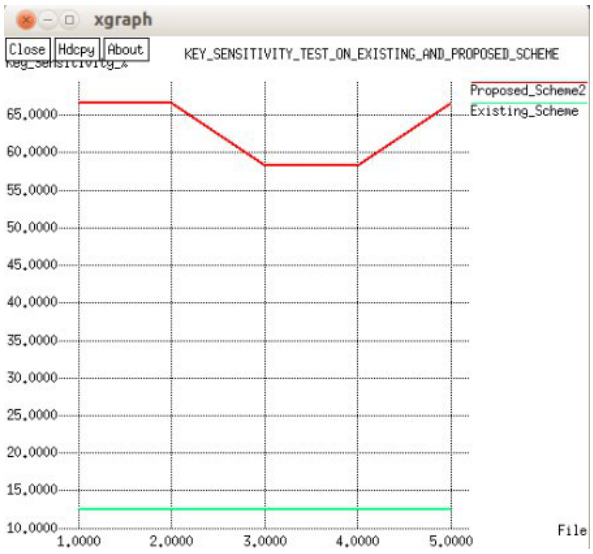

Figure 7: Key Sensitivity Percentage %

7 Conclusions and future scope

As seen from the above results, the proposed scheme provides diffusion and confusion, hence ensures the occurrence of the avalanche effect. The percentage of plaintext sensitivity and key sensitivity in the proposed scheme is better than genetic approach and RC5 therefore it is secure. In future, more algorithms can be implemented in Wireless Sensor Networks and compared with the proposed scheme.

References

[1] Harmanjit Kaur, "International Journal of Advanced Research in Computer Science and Software Engineering", Volume 5, Issue 4, April 2015, IJARCSSE-2015, p. 1495-1501.

[2] Shailesh N. Sisat, Shrikant J. Honade, "Review on Security and Privacy in Wireless Sensor Network", Volume 2, Issue 3, March 2014, IJARCSSE-2014, p. 21-26.

[3] Kamanashis Biswas, Vallipuram Muthukkumarasamy, Elankayer Sithirasenan, and Kalvinder Singh, "A Simple Lightweight Encryption Scheme for Wireless Sensor Networks", ICDCN, Springer-2014, p.499-504

[4] Kamanashis Biswas, Vallipuram Muthukkumarasamy, Kalvinder Singh, "An Encryption Scheme Using Chaotic Map and Genetic Operations for Wireless Sensor Networks", IEEE sensors journal, IEEE-2014, p. 581-588.

[5] Nivedita Mukherjee, "A Dynamic Cryptographic Algorithm To Provide Nodal Level Security In Wireless Sensor Network", IEEE-2010, p.189-194.

[6] V. Thiruppathy Kesavan, S. Radhakrishnan, "Secret Key Cryptography based Security Approach for Wireless Sensor Networks", IEEE-2012, p.185-191.

[7] Nikos Komninos, Hamed Soroush, and Mastooreh Salajegheh, "LEE: Light Weight Energy Efficient Encryption Algorithm for Sensor Networks", IEEE-2012, p.493-498.

[8] Gulshan Kumar, Mritunjay Rai, and Gang-soo Lee, "An Approach to Provide Security in Wireless Sensor Network Using Block Mode of Cipher", Springer-2011, p.101-112.

[9] Teng Zhang, Tingyuan Nie, "A Study of DES and Blowfish Encryption Algorithm", TENCON, IEEE-2009, p.1-4.

[10] Soufiene Ben Othman, Abdelbasset Trad, Habib Youssef, "Performance Evaluation Of Encryption Algorithm For Wireless Sensor Networks", ICITS, IEEE-2012, p.680-687.

[11] Xueying Zhang, Howard M. Heys, Cheng Li, "Energy Efficiency of Symmetric Key Cryptographic Algorithms in Wireless Sensor Networks", IEEE-2010, p.168-172.

[12] Santar Pal Singh, S.C. Sharma, "A Survey on Cluster Based Routing Protocols in Wireless sensor Networks", ICACTA, Elsevier-2015, p.687-695.

[13] G. R. Sakthidharan and S. Chitra, "A survey on wireless sensor network: An application perspective", ICCCI, p. 1–5, 2012.

[14] M. Wang, "Differential Cryptanalysis of PRESENT", In proc. of Africa crypt 2008, to appear, Cryptology e-Print Archive: Report 2007/408.

[15] S. Lucks, "Ciphers Secure against Related-Key Attacks", Proc. of Fast Software Encryption, LNCS 3017, 2004, pp. 359-370.

[16] Canteaut, "A Linear Feedback Shift Register, In Encyclopedia of Cryptography and Security, pp. 355-358. Springer (2002).

[17] S. Knellwolf, W. Meier, and M. Naya-Plasencia, "Conditional differential crypt-analysis of NLFSR-based crypto-systems, ASIACRYPT", LNCS, vol. 6477, pp. 130–145, 2010.

[18] S. Knellwolf, W. Meier, and M. Naya-Plasencia, "Conditional differential crypt-analysis of trivium and katan, Selected Areas in Cryptography" (SAC-2011), LNCS, vol. 7118, 200–212, 2011.

[19] L. Wei, C. Rechberger, J. Guo, H. Wu, H. Wang, and S. Ling, "Improved meet-in-the-middle cryptanalysis of ktantan, Information Security and Privacy", LNCS, vol. 6812, 433–438, 2011.

[20] T. Suzaki, K. Minematsu, S. Morioka and E. Kobayashi. "TWINE: A lightweight block cipher for multiple platforms", SAC, LNCS, 2012.