Indian Journal of Science and Technology, *Vol 8(S9), 198–206, May 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Implementation of Modified RC4 Algorithm for Wireless Sensor Networks on CC2431

B. Kiruthika[1*], R. Ezhilarasie[2] and A. Umamakeswari[3]

[1]Embeddd Systems, School of Computing, SASTRA University, Tirumalaisamudram, Thanjavur - 613401, Tamil Nadu, India; bkiruthika27@gmail.com

[2]School of Computing, SASTRA University, Tirumalaisamudram, Thanjavur - 613401, Tamil Nadu, India; ezhil@cse.sastra.edu, aum@cse.sastra.edu

Abstract

Advancements in wireless communication and computing devices have made Wireless Sensor Network (WSN) interesting. Since WSN is deployed in hostile environments, the security becomes a common concern. WSN demands for lightweight security schemes due to the fact that the nodes in the networks are resource constrained. In order to provide secure data transmission in wireless sensor networks, cryptographic algorithms has to be incorporated. RC4 is one of the commonly used stream cipher due to its speed and simplicity in implementation. The drawback of RC4 is weakness in the permutation of internal state table during initial phase. To overcome the limitations in existing RC4, RC4 stream cipher with modified state table is proposed. The system reduces the known outputs of the internal state and also improves the execution speed. Modification is made on computational steps of the initialisation phase. The modified method tends to provide high degree of randomness in the permutation of the state variable than the original RC4. The algorithm design is targeted to encrypt/decrypt 8-bit ADC reading of TICC2431.

Keywords: CC2431, RC4, Security, Stream Cipher, Wireless Sensor Networks (WSN)

1. Introduction

Advancements in wireless communication and computing devices have made Wireless Sensor Network appealing. Wireless sensor networks[1] are composed of resource constrained nodes that are assigned with multiple functions, such as monitoring and processing the data, in order to serve for various application purposes. The key components of these networks are the sensor nodes and the base stations. These networks are either of homogenous or heterogeneous type that are deployed at pre-determined locations or randomly positioned using various techniques. Sensor nodes are fixed, however mobile nodes can be deployed based on the requirements of the application. These nodes are programmed to communicate via wireless medium. The nodes being resource constrained, has limitation in their physical size, and has restricted on-board resources[2]: memory requirements, power resources, processing capability and bandwidth for communication. Primarily, WSN are mainly utilized for military surveillance later they were extended to serve for commercial applications such as medical and environment monitoring, machinery monitoring in manufacturing applications, home automation, traffic control etc. The sensor nodes in these environments are subject to periodical surveillance. In such hostile environments, the security of WSN becomes a common concern. For instance, consider a sensor node serving for military application; it is essential that it retains the gathered information secret and reliable. Information leakage[3] may result in breaking the privacy of the environment. Additionally, this also eases for eavesdropping

**Author for correspondence*

and packet injection in sensor network. The demand for securing the sensor networks at the design time to facilitate secured functioning of the network which involves sensitive data requires considering the issues mentioned above. However, affording security for wireless networks is tedious as the traditional security schemes may not be compatible for it. WSN demands for lightweight security[4] schemes so that the overhead imposed can be reduced which in turn does not affect the overall performance of the network. This is due to the fact that the network is resource[5] constrained. These design constraints pose various security issues and makes the network susceptible to several attacks. The identified key areas that are to be considered for providing security to sensor networks are: Cryptography, Key management, Secured routing, Secured location, Secured data fusion, Attack detection and prevention and other issues in security. In order to achieve high level of security in WSN, cryptographic operations such as encryption, authentication and so on, has to be performed. The selection of appropriate mechanism is necessary in securing the network. The choice of ciphers depends on various factors: size of operands in encryption, operating modes, considerations for security, key stream generation key selection, energy efficiency of the network. This paper focuses on enhancing the security of the WSN by applying cryptographic techniques. One of the commonly used cryptographic algorithms in WSN is Rivest Cipher 4 (RC4). RC4[6] is a stream cipher that can be applied to various applications. It poses weakness including a correlation problem between the internal states. Hence a new modified RC4 with two state tables as an enhancement to RC4. The modification is made in the permutation of key stream generation. The design is targeted to ZigBee TICC2431 SoC.

This paper is organised as: Section II emphasis an overview of Stream Ciphers. Section III describes the basic framework of RC4 algorithm. Section IV illustrates the round operation of the proposed RC4.The proposed method of hardware implementation of RC4 algorithm is explained in Section V. Section VI demonstrates the results and the conclusion is drawn in Section VII.

2. Related Work

With numerous cryptographic techniques[7] available only few among them may be applicable in resource constrained environments like WSN. In order to ensure high level of security in these environments Lightweight cryptography can be applied. As the traditional algorithms requires high amount of resources, these crypto techniques has the ability to provide security exploring minimal processing capabilities of the nodes. Symmetric and asymmetric cryptography[6] are the two widely used methods. Asymmetric cryptography, which can also be termed as Public Key Cryptography, involves usage of two secret keys: a key for encryption- public key and another key for decryption-private key. Although this method tends to provide high secrecy of the data, it may not be suitable for the resource constrained devices as it requires parallelisation, unrolling and pipelining which may not be effectively achieved. Whereas Symmetric ciphers involves usage of a single key for encryption as well as for decryption. This key is maintained secret between the sender and the receiver. Symmetric ciphers consumes less mathematics and this method does not suffers problem as that of as asymmetric algorithms and can be easily implemented in Wireless sensor networks. The two variants of symmetric cipher implementations are: Block and Stream ciphers.

Block ciphers[8] has a message of fixed length known as blocks, which is of requires enormous computation and substitution of key to encrypt the block of data. Fiestel Network and Substitution Permutation Network (SPN) are the two mainly used structures in block ciphers. Apart from the structure, block ciphers has several modes of operation: Output Feedback mode, Counter mode (CTR) and Cipher Feedback mode (CFB).

Stream ciphers[9] has a variable message length, which requires a substitution table for encryption. It consists of two functions: a state update function and an output function. The state function involves continuous permutation and updating of the table for encryption. The key-stream bits are obtained from the output function used for encryption using a bit-wise XOR operation. Based on the state functions used, stream ciphers can be classified as synchronous stream ciphers independent of the preceding cipher bit and asynchronous stream ciphers which depends on the preceding cipher bit.

Symmetric ciphers[9], being an idle choice for WSN, further selection of algorithm has to be made based on the factors such as: size of the operands, considerations for security, operating modes, and key stream setup/ expansion. WSN which involves the transmission of data that is gathered from other nodes as a result of continuous

monitoring in the deployed environment and as the sensor nodes has limited energy and processing capabilities, stream ciphers are more appropriate than block ciphers. As stream ciphers deals with small bits of information rather than blocks of data, they are smaller and faster in comparison with block ciphers. This does not result in any noise in transmission of data as the data is not carried over in chunks providing high amount of efficiency. Stream ciphers are suitable for networks that involve handling of unknown data or continuous transmission of data.

RC4 is a stream ciphers[5] that is widely used and is often implemented in software. Although various other stream ciphers that are more secure and efficient than RC4 have been defined, still it the most common algorithm. It is mainly due to its speed and simplicity in implementation. The cipher withstands several cryptanalysis attempts and proves to be secure. In this article, we analyse the basic framework of RC4 and present a hardware implementation of modified RC4 algorithm that generates the key stream at faster rate than the original RC4.

3. RC4 Algorithm

RC4, a synchronous stream cipher[10–12] was formally proposed by Ron Rivest in 1987 for RSA Data Security. This algorithm satisfies for both security and efficiency for lightweight algorithms, suitable for resource constrained environments. A XOR operation is performed on the data and the key stream. The generated key stream, as result of the algorithm is independent of the plaintext. An N bit-state table is initialised using a key of variable length ranging from 1 to N. This state table generates subsequent pseudo-random bits which are used to produce a pseudo-random key stream. The resulting stream is XOR-ed along with the plain text to obtain the cipher text. The algorithm used for decryption is same as that of the one used for encryption.

The algorithm is composed of two phases, such as the Key Scheduling Algorithm (KSA) and the Pseudo-Random Generation Algorithm (PRGA).

The KSA is the initialization stage, the N-bit state table S is permuted using the key K. The KSA can be summarized as pseudo code:

The above procedure swaps the position of the bits present in the state table S. These values are used as input for the PRGA. This uses pseudorandom permutation to

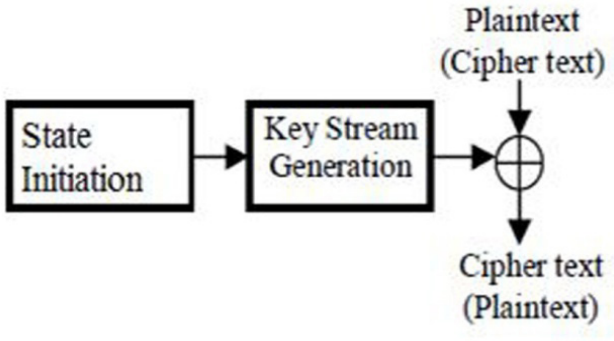

Figure 1. Functional steps in RC4

generate stream of pseudorandom values. PRGA operation can be summarized as pseudo code:

At the sender, the output stream Z is XOR-ed with input stream to obtain the cipher text $(C = M \oplus Z)$, where M is the input message with N bits. At the receiver, the output stream Z is XOR-ed back with cipher text to obtain the plain text $(M = C \oplus Z)$.

Several weaknesses in RC4 have been identified. One of the identified problems of RC4 is the weakness of the KSA. The state table permuted along with the weak keys can be cracked easily. In order to strengthen the KSA, a modified method that generates the state table of KSA in different mechanism is proposed. The steps involved in the modified RC4 structure and the pseudo code is described in section IV.

1. Procedure for KSA (Secret Key K)
2. Initialize: The value of index j = 0
3. For: i = 0 to N-1;
4. Initialise: S[i] ← i;
5. Increment: j ← (j + S[i] + K[i]) mod N;
6. Swap: S[i] ↔ S[j];
7. Return: S
8. End Procedure

Figure 2. Algorithm I. Key Scheduling Algorithm (KSA).

1. Procedure for PRGA (S)
2. Initialize: The value of indices: i = 0, j = 0;
3. While TRUE do
4. Increment: i ← (i + 1) mod N;
5. Increment: j ← (j + S[i]) mod N;
6. Swap: S[i] ↔ S[j]
7. Output: Z= S [(S[i] + S[j]) mod N]
8. End While
9. End Procedure

Figure 3. Algorithm II. Pseudo Random Generation Algorithm (PRGA).

4. Modified RC4 Algorithm

As mentioned in the previous section, RC4 has substantial number of weakness in the KSA phase. In this section, we propose a RC4 stream cipher with modified state table as an enhancement to RC4. It is one of the families of RC4 stream cipher to reduce known outputs of the internal state and also improving the execution speed. The proposed method includes the same phases as that of the original algorithm. Modification is made on computational steps of the initialisation phase.

The KSA consists of two state variables S_1 and S_2. S_1 is assigned from 0 to (N/2)-1 and S_2 is filled with remaining N/2 numbers from N/2 to N-1. The key K of length L is used as a seed for the states S_1 and S_2. Therefore S_1 and S_2 are used as two secret random inputs for second phase. The KSA can be summarized as pseudo code:

The PRGA phase produces two keys Z_1 and Z_2 in each cycle instead of one as with the original RC4. The swapping between the state variables is defined by three indices: i, j_1, j_2. The PRGA can be summarized as pseudo code:

At the sender, the output streams Z_1 and Z_2 are XOR-ed with the input stream to obtain the cipher text $(C = M \oplus Z_1 \oplus Z_2)$, where M is the input message with N bits. At the receiver, the out streams Z_1 and Z_2 are XOR-ed back with cipher text to obtain back the plain text $(M = C \oplus Z_1 \oplus Z_2)$. The analysis of the modified approach is discussed in Section V.

5. Analysis of Modified RC4

To analyse the key stream of the modified RC4, the steps involved in original RC4 is exploited. The Key Scheduling phase aims at producing a pseudorandom permutation of elements from a state table S. In the next phase, PRGA produces an output byte from the random position of state table S in each cycle. In each cycle, the values of state table S are obtained by swapping the position of elements. This swapping is performed between two values pointed at one known position and the other denoted by an index value. Security of the RC4 depends on following aspects:

- Permutation of variables in state table S during the initialisation.
- Value of index pointer j.
- The output value Z which is obtained using index pointer.

The factors mentioned are essential but aren't adequate. KSA performs permutation of variables in the state table S using the secret key K. The index j is incremented by $j = (j + S[i] + K[i]) mod\ N$. As the variables of the state table S are distributed uniformly, the value of j is also obtained so.

The internal state of RC4 includes a state table S and an index pointer j. The function performs an update on the variable j and the permutation of state table S using a swap function $Swap(S[i], S[j])$. Selections of the position of the values to be swapped are done uniformly as j. This enhances the internal state of the algorithm and is computed distributed.

Let us study the modified RC4. The first variation from the original algorithm is in permutation of N elements in the state table S. In the modified algorithm the state array contains two variables S_1 and S_2 each of length $\frac{N}{2}$. The permutation is performed on these variables. Assuming that the permutation of N elements in both S_1 and S_2 are uniformly distributed with key K as seed. The index pointer j is incremented by two statements in each loop for S_1 and S_2.

1. Procedure for KSA (Secret Key K)
2. Initialise: The index i
3. For: i = 0 to N/2-1;
4. Initialise: $S_1[i] \leftarrow i$;
5. For: i = N/2 to N-1;
6. Initialise: $S_2[i-N/2] \leftarrow i$;
7. Initialise: The value of index j=0;
8. For: i = 0 to N/2-1;
9. Increment: $j \leftarrow (j + S_1[(i + K [i \bmod L]) \bmod N/2 + K [i \bmod L]) \bmod N/2$;
10. Swap: $S_1[i] \leftrightarrow S_1[j]$;
11. Initialise: The value of index j=0;
12. For: i = 0 to (N/2-1);
13. Increment: $j \leftarrow (j + S_2[(i + K [i \bmod L]) \bmod N/2 + K [i \bmod L]) \bmod N/2$;
14. Swap: $S_2[i] \leftrightarrow S_1[j]$;
15. Return: S_1 and S_2
16. End procedure

Figure 4. Algorithm III. Modified Key Scheduling Algorithm (M-KSA).

1. Procedure for PRGA (S)
2. Initialize: The value of the indices: $i = 0, j_1 = j_2 = 0$;
3. For i=0 to L/2;
4. Increment: $i \leftarrow (i + 1) \bmod N/2$;
5. Increment: $j_1 \leftarrow (j_1 + S_1[i]) \bmod N/2$;
6. Swap: $S_1[i] \leftrightarrow S_2[j_1]$;
7. Output $Z_1 = S_1 [(S_1[i] + S_1[j_1]) \bmod N/2]$;
8. Increment: $j_2 \leftarrow (j_2 + S_2[i]) \bmod N/2$;
9. Swap: $S_2[i] \leftrightarrow S_1[j_2]$;
10. Output: $Z_2 = S_2 [(S_2[i] + S_2 [j_2]) \bmod N/2]$;
11. Return: Z_1, Z_2
12. End procedure

Figure 5. Algorithm IV. Modified Pseudo Random Generation Algorithm (M-PRGA).

$$j = (j + S_1 1\,[(i + K[i \bmod L])\bmod N / 2 + K\,[i \bmod L])\ \bmod N/2$$

and

$$j = (j + S_1 2\,[(i + K[i \bmod L])\bmod N / 2 + K\,[i \bmod L])\ \bmod N/2$$

Consider PRGA, in original RC4 a single index pointer is used to update the state variable S, $j = (j + S[i])mod\ N$. A swapping operation is performed on the resulting state array before the output $Z = S\left[(S[i] + S[j])mod\ N\right]$ is obtained. This produces a single key stream from the state variable S. In case of the modified algorithm, it requires two index pointers j_1 and j_2. The swapping operation is performed on the state variables S_1 and S_2 with different index pointers which has high randomness than the original RC4. This implies that the values of the indices from which the outputs Z_1 and Z_2 are taken are also uniformly distributed.

The modified method is faster than RC4 as it requires five modulo functions and two swaps to generate two key streams for each iteration in the PRGA phase whereas RC4 stream cipher requires only three modulo functions and one swap to generate one key stream. This method merges cumulative randomness of the initial states with permutation of the two state tables in key generation to solve the known outputs. The modified method tends to provide high degree of randomness in the permutation of the state variable than the original RC4.

The performance of modified RC4 and RC4 based on the randomness of resultant key. Randomness measures the probabilistic outcome of the output that is uniformly distributed and unpredictable. The key features of randomness are: uniform distribution, unpredictable, untraceable and capability to overcome correlation attack. Although randomness provides high assurance in random generator, it does not provide an absolute trust.

Figure 6 Illustrates the randomness of key in modified RC4 and original algorithm. On the basis of bytes obtained from PRGA the graph is plotted. The value on the X-axis denotes the key length ranging from 0-32 bytes whereas the Y-axis represent the key obtained from PRGA function, ranging from 0 to 255. It can be observed from the graph that the points are spread randomly and are distributed evenly. This assures that the keys generated in Modified method is untraceable than the original RC4 and accounts for correlation attack.

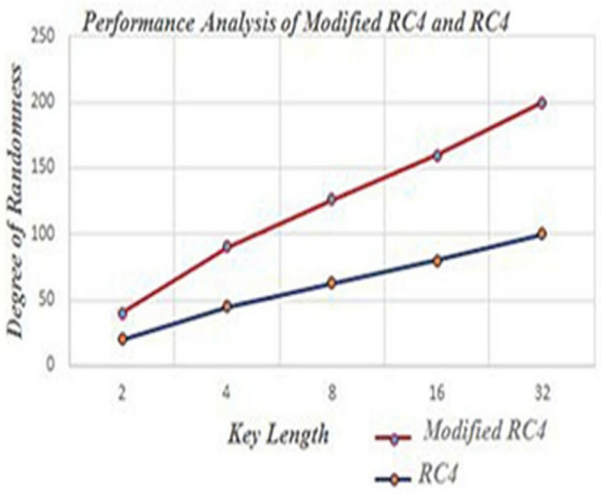

Figure 6. Performance analysis of modified RC4 Vs RC4.

6. Implementation

The proposed modified RC4 algorithm was implemented into WSN using ZigBee CC2431. The platform that was used in this application is IAR Workbench EW51 and the radio frequency protocol is based on IEEE 802.15.4 standard. Few main features of the hardware are shown in Table 1.

To implement the modified RC4 algorithm, a CC2431 node is configured as an end device and the other node is configured as coordinator. The Coordinator is connected to a PC via UART to display the received values.

The on-board 8-bit ADC value of the end device is encrypted using the algorithm and is transmitted using RF module to the coordinator. At the coordinator, the received value is decrypted and is displayed in PC via UART communication. The snapshot of the system setup is shown in Figure 7.

7. Evaluation and Results

The module that performs the encryption and decryption requires about 300 lines of code. The code for encryption and decryption was compiled and downloaded into the hardware. The encryption algorithm was downloaded in the end device. The value of the 8-bit ADC was encrypted with an 8-bit key using the modified algorithm and the decryption is done by the coordinator.

Table 1. Specifications of CC2431

ZigBee CC2431 SoC		
MCU	8051	
	Model	8051F320
	Type	8-Bit RISC
	Programmable Flash	128 KB
	RAM	8KB
Radio	CC2420	
	Frequency	2.4 GHz
	Data Range	250 kbps
	PC Interface	USB

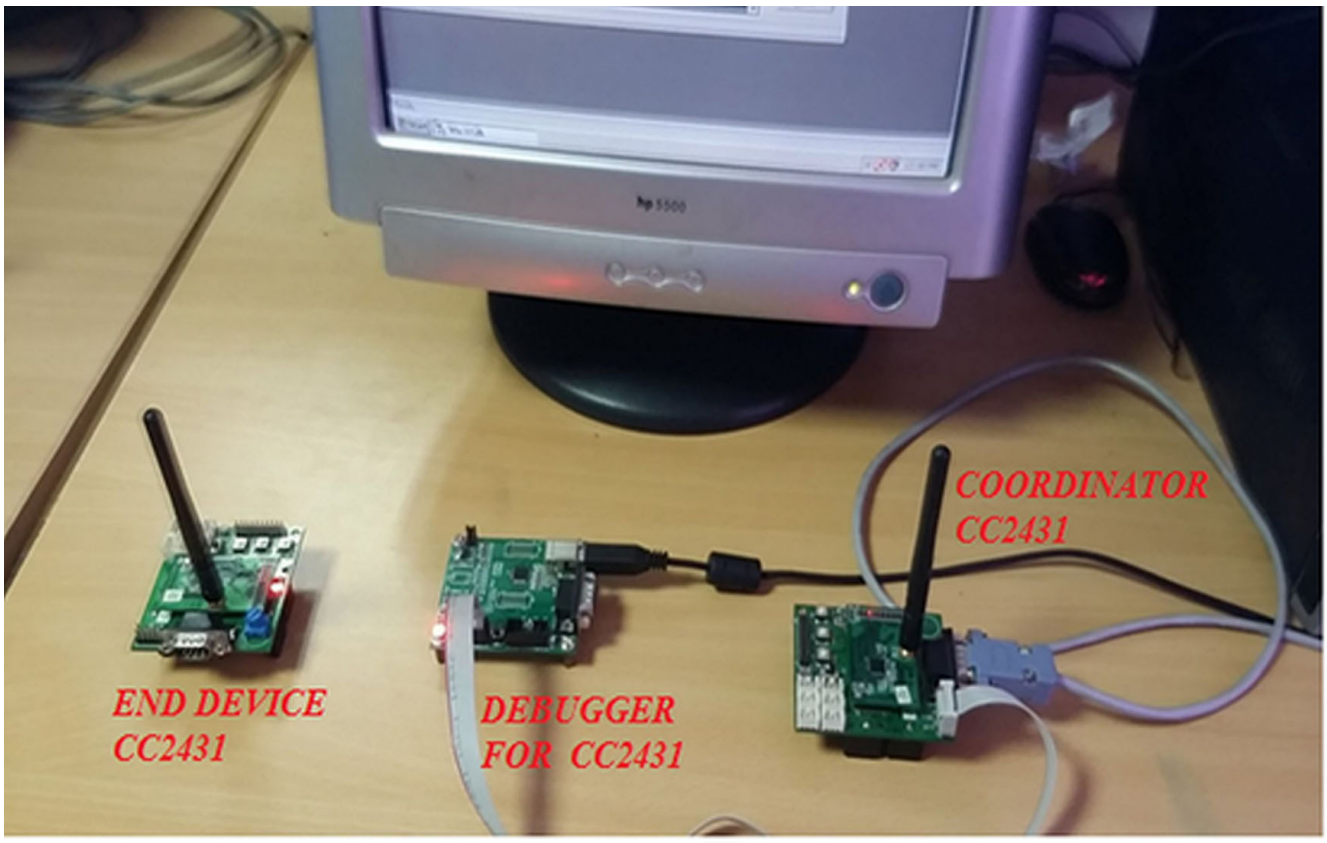

Figure 7. System Setup.

In order to verify the encryption steps, the encryption algorithm was downloaded to the hardware. The encrypted value was transmitted to the coordinator from the end device. The values received at the coordinator were viewed through hyper terminal in the PC connected to the coordinator via UART. The snapshot of the values received at the coordinator is shown in Figure 8.

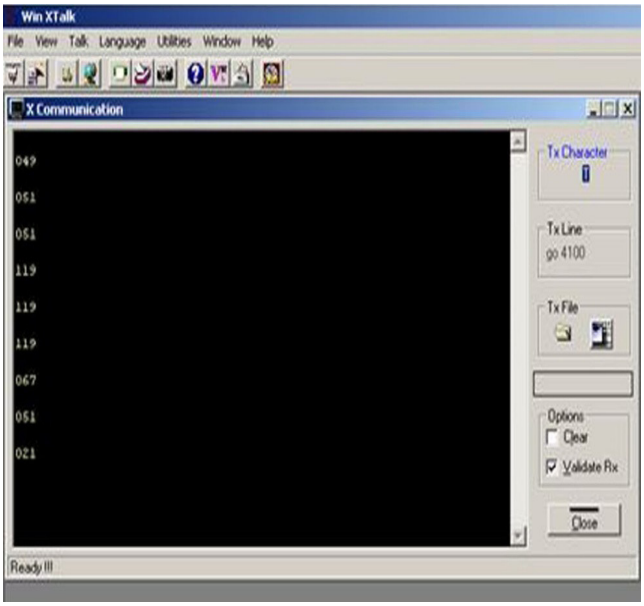

Figure 8. Reading after encryption at End device.

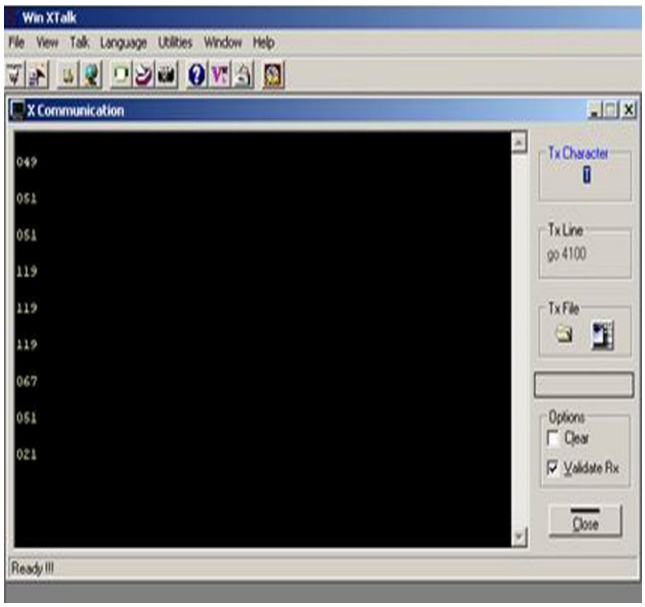

Figure 9. Reading after decryption at Coordinator.

The decryption algorithm was downloaded in the coordinator and the original readings of the 8-bit ADC were obtained. The snapshots of the readings are shown in Figure 9.

The original RC4 and modified RC4 algorithms were implemented on PC and the results shows that the modified algorithm is faster than the original algorithm. The key stream is faster than the original algorithm. This is mainly due to the reason that randomness of the initial state is improved from the original algorithm.

8. Conclusion

RC4 stream cipher is an efficient cryptographic technique for securing the data in a communication channel as it is simpler and faster than many other algorithms. A modified RC4 algorithm with variation in state table computation was proposed to enhance the randomness in the key stream generation process. The key generation time was found to be faster than the original RC4 algorithm. The proposed design was implemented into WSN using TICC2431 and the results were found to be efficient than the original RC4. Several security tests can be performed on the proposed algorithm to test the strength of the generated key stream for wireless sensor networks as a future work.

9. Acknowledgement

This work was supported by FIST programme, Department of Science and Technology India, Grant no: SR/FSC/ETI-371/2014 at SASTRA University, Tirumalaisamudram, Thanjavur.

10. References

1. Kavitha T, Sridharan D. Security vulnerabilities in wireless sensor networks: A survey. Journal of Information Assurance and Security. 2010; 5(1):31–44.
2. Ashok J, Thirumoorthy P. Design considerations for implementing an optimal battery management system of a wireless sensor node. Indian Journal of Science and Technology. 2014; 7(9):1255–9.
3. Panda M. Security in wireless sensor networks using cryptographic techniques. AJER. 2014; 3:50–6.
4. Kong JH, Ang L-M, Seng KP. A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments. J Netw Comput Appl. 2015; 49:15–50.
5. Sheeba TB, Rangarajan P. Area efficient cryptographic ciphers for resource constrained devices. Life Science Journal. 2013; 10(3):1107–14.
6. Sasi SB, Dixon D, Wilson J, No P. A general comparison of symmetric and asymmetric cryptosystems for WSNs and an overview of location based encryption technique

for improving security. IOSR Journal of Engineering. 2014; 4(3):1.
7. Kumar SA, Ilango P. Data funnelling in wireless sensor networks: a comparative study. Indian Journal of Science and Technology. 2015; 8(5):472–80.
8. Law YW, Doumen J, Hartel P. Benchmarking block ciphers for wireless sensor networks. 2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems; IEEE. 2004.
9. Minglin Y, Junshuang M. Stream ciphers on wireless sensor networks. 2011 Third International Conference on Measuring Technology and Mechatronics Automation (ICMTMA); IEEE; 2011.
10. Mousa A, Hamad A. Evaluation of the RC4 algorithm for data encryption. IJCSA. 2006; 3(2):44–56.
11. Kitsos P, Kostopoulos G, Sklavos N, Koufopavlou O. Hardware implementation of the RC4 stream cipher. 2003 IEEE 46th Midwest Symposium on Circuits and Systems; IEEE; 2003.
12. Pu C-C, Chung W-Y. Group key update method for improving RC4 stream cipher in wireless sensor networks. IEEE International Conference on Convergence Information Technology; 2007.