ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

Implementation of AES algorithm on ARM processor for wireless network

Vinayak Bajirao Patil [1], Prof.Dr.Uttam.L.Bombale [2] ,Pallavi Hemant Dixit.[3]

M.Tech, Department of technology, Shivaji University, Kolhapur, India [1]

Professor, Department of technology, Shivaji University, Kolhapur, India [2]

M.Tech, Department of technology, Shivaji University, Kolhapur, India [3]

Abstract: Wireless network can offer businesses more flexible and inexpensive ways to send and receive data. The wireless network is useful to increase efficiency, greater flexibility and mobility for users. Security is major concern in wireless networks. As we share the data through wireless network it should provide data confidentiality, integrity and authentication. The Advanced Encryption Standard (AES) algorithm has popular; as it is implemented into embedded system and it is best solution for security.

This paper presents wireless network which contain both mobile nodes. Here the implementation of AES algorithm on hardware in combination with part of software on both nodes. Realization of AES algorithm on ARM processor with minimum memory will be useful for deploying it in low cost applications. Algorithm is compiled using KEIL compiler targeting into an ARM LPC2148 processor board. The results are presented into this paper for key length of 128 bits. After AES encryption the cipher text is send through GSM module to another node, while receiving cipher text through GSM module the plain text is obtained by AES decryption.

Keywords**:** Wireless Network, AES, Embedded system, GSM

I. Introduction

It is easier to apply cryptographic solutions on computer based communication systems than on conventional systems. It is not feasible to dedicate a general computer for each of such systems. Instead, a cheap and portable embedded system can be developed to ensure the communication security. Embedded systems can be powered by microcontroller, DSP, or ASIC. Microcontroller based embedded systems have lowest cost, which is one of the basic criteria of an embedded system design. Variety of microcontrollers available, each have different processor and peripheral devices inside them. ARM7TDMI is a popular embedded processor that has a lion's share of the market. It is reliable, has low cost, low power consumption and small physical size [1].

Rijndael is a symmetric block cipher of variable key and block size. Rijndael invented by two Belgian cryptographers Vincent Rijmen and Joan Daemen. Since it has won the Advanced Encryption Standard compilation by National Institute of Standards and Technology (NIST) in 2001, it is called as AES. The selection of AES was done based upon the parameters such as security, performance, efficiency, flexibility, and implementability by NIST. The selected algorithm, viz., Advanced Encryption Standard (AES), has replaced DES and published as FIPS 197 in November 2001[2].

Hardware and software implementation of AES is challenging. Hardware implementation is common in high speed application. Software implementation [3] is relatively slow and consumes processor time and hence for embedded system application hardware implementation is popularly done. AES is implemented in various hardware platforms based upon controllers / processors with 8-bit words to 64-bit words. The AES implementation on the 8-bit controllers is easy with low cost and used for low end applications, whereas large bit processors are expensive and used for high end applications. In this paper we implemented AES algorithm on ARM LPC2148. The compiler tool KEIL is used. The AES code is written in embedded c. The time required for execution of Key expansion for different rounds of AES is calculated also the time required for encryption and decryption is calculated.

II. Related Work

Rijndael has been open for research since it has been accepted as AES. Different papers have been published on aspects of the algorithm, like its strength, efficiency, software and hardware implementation etc… This chapter summarized different papers on AES implementation optimization for varies platforms.

K.Atasu et al [9] have optimized AES implementation for speed and memory efficiency. The speed optimization focuses on the linear mixing module called MixColumn.. In this paper, they propose a new approach that combines the two approaches. It uses the standard approach [2] for the

encryption and the transposed approach [8] for the decryption. This new combined approach has a better performance than the pure standard and the pure transposed approaches.
Kim and Verbauw [9] have optimized Rijndael implementation on 8-bit AVR microcontroller. The performance improvement has come from holding the state matrix values and arithmetic logic operation operands on registers. In their implementation, they have tried to minimize memory access as much as possible. Direct multiplication is used instead of table lookup for mix column. Therefore, in their implementation memory will be referenced only for round key and S-Boxes, which in this case is impossible to store in registers.
Ashruf et al in [7] have implemented AES in Molen hardware. Molen processors are a combination of General Purpose Processors (GPP) and Field Programmable Gate Array (FPGA). Except its high cost that depends on its size, FPGA has the highest speed of all processors and highly flexible. The result shows that Molen architecture implementation runs as fast as pure FPGA implementation. Besides, since the amount of FPGA required for this implementation is smaller than the pure FPGA implementation, the const is also low.
T.Ravichandra Babu [11] have implemented AES onto ARM platform. Hardware and software implementation of AES has done. The AddRoundKey operation and Mix Column operation is implemented on hardware. ShiftRows and Substitute byte is implemented on software.

III. AES ALGORITHM

All Rijndael was designed to have the following characteristics:
• Resistance against all known attacks.
• Speed and code compactness on a wide range of platforms.
• Design Simplicity.

AES is a symmetric block cipher with block length of 128 bits. It allows three different key lengths 128,192 and 256 bits. In encryption process; for processing of 128 bit keys required 10 rounds, 192 bit keys required 12 rounds and 256 bit keys required 14 rounds. AES is a round based algorithm. For encryption and decryption ;each round has four functions excepting last round (Last round required three functions).The encryption algorithm has four round functions SubByte, ShiftRows, MixColumn(Omitted in last round) and AddRoundKey. The decryption also has the same number of rounds with reverse transformation, order of round function is different i.e. InvShiftRow, InvSubByte, AddRoundKey and InvMixColumn (Omitted in last round) [3][4]. TABLE I shows number of AES parameters for the accepted three AES versions.

TABLE I
AES PARAMETERS

	AES-128	**AES-192**	**AES-256**
Key Size (Bits)	128	192	256
Plaintext box size (Bits)	128	128	128
Number of rounds	10	12	14

The key expansion algorithm generates 128 bit key for each round and one more key for initial AddRoundKey function. The same expanded key is used for encryption and decryption except for decryption it reads in reverse order.
Rijndael is a very good performer in both hardware and software across a wide range of computing environments regardless of its use in feedback or non-feedback modes [6]. Its key setup time is excellent, and its key agility is good. Rijndael's very low memory requirements make it very well suited for restricted-space environments. Rijndael's operations are among the easiest to defend against power and timing attacks.

The input bit sequence is first transformed into byte sequence. In next step a two-dimensional array of bytes (called the State) is built. The State array consists of four rows of bytes, each containing 4 bytes. All internal operations (Cipher and Inverse Cipher) of the AES algorithms are performed on the State array. The AES algorithm basically consists of four byte oriented transformation for encryption and inverse transformation for decryption process namely,

a) Byte substitution using substitution box table (S-box)

b) Shifting rows of the state array using different offsets. (Row transformation)

c) Mixing the data within each column of the state array. (Mixing columns)

d) Adding a round key to the state. (Add round key)
The fig.1 shows AES Rijndael Encryption and Decryption structure, where the input to the encryption and decryption algorithm is 128 bit block. The key provided is expanded into an array of forty - four 32 bit words, w[i]. Four distinct words forming 128 bits serve as a key for each round in both encryption and decryption. In encryption process first four words w(0 – 3) are used as key in the first round but for decryption last four words w(40 – 43) are used in the first round.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

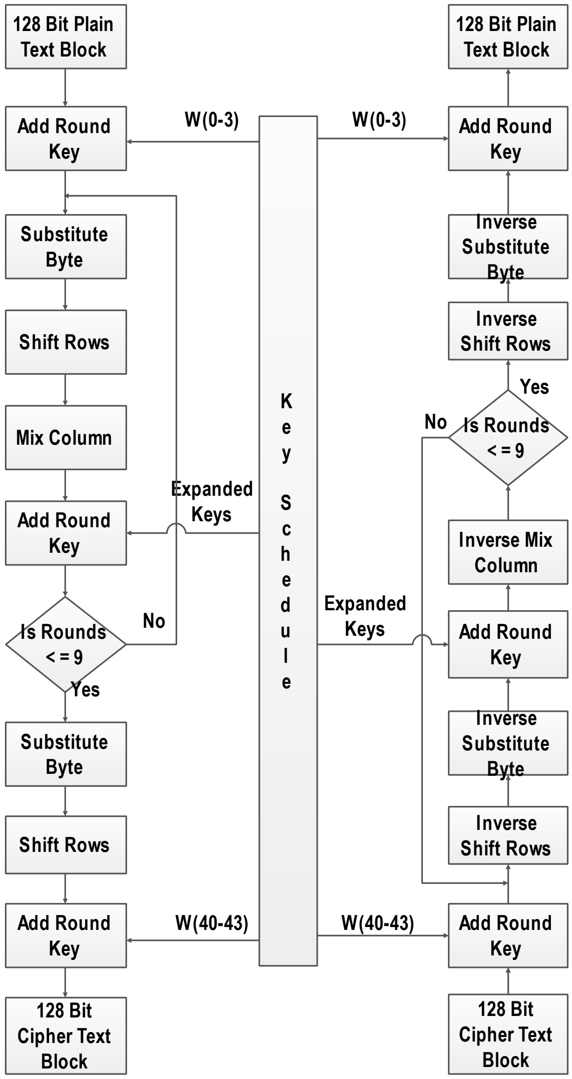


Fig. 1 AES Algorithm

IV. GSM SYSTEM

Global System for Mobile communication (GSM) is one of the best trustable wireless communication systems that can be accessed and used very easily. It is cost effective either if we consider the price of its transceiver module or the subscription fees. With the trend of huge growing usage of GSM during the past decade, network services is extended beyond speech communication to so many other custom specified applications, machine automation and machine to-machine communication. In order to choose the proper technology for communication infrastructure, in Fig.2 a comparison between available wireless communication technologies was made [5].Our demand is related to high covering range and high data rate. From following graph the UWB technology (Ultra Wide Band) is designed for PANs (Personal Area Network) and has a covering range of 10m, it also loses its viability in the mobile network field. Finally we remain only with two options for communication infrastructure technology, ZigBee (practically it was especially develop for such application a mobile networks), and GSM through its lower cost data transfer services GPRS, EDGE and the newly entered in the cheap communication category, HSDPA.

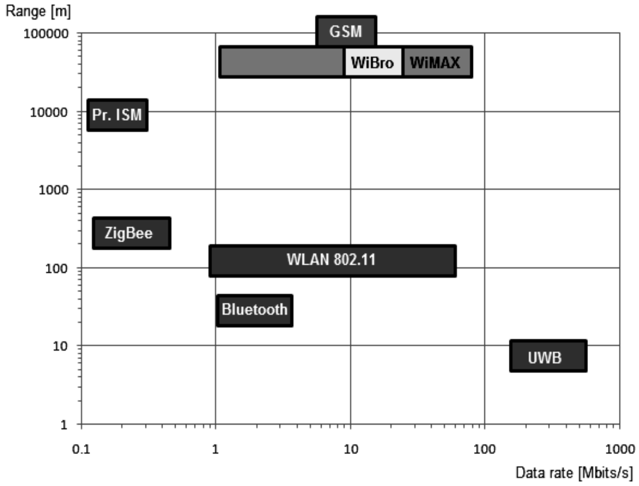


Fig. 2 Graphical representation of performances of wireless Technologies

V. IMPLEMENTATION OF SYSTEM

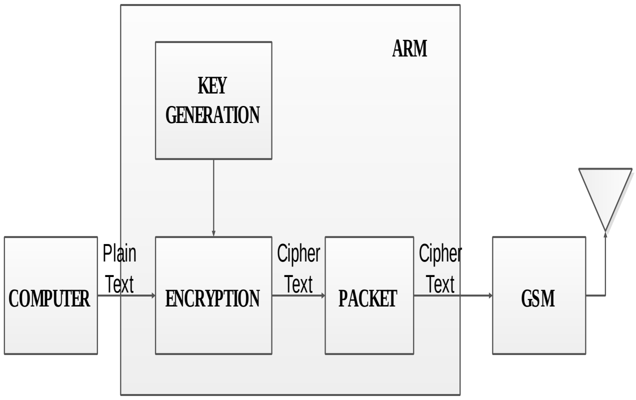


Fig. 3 Node A

Fig.3 and Fig.4 shows the block diagram of Node A and Node B respectively. In Node A, computer is connected to UART 0 of ARM LPC2148 board. The AES Encryption code is dumped onto ARM board. When the plain text is given from computer to ARM then AES encryption algorithm converts that plain text into cipher text. Key generation generates keys which used in encryption process. Further the cipher text is given to GSM module. The cipher text is send through GSM module to Node B.

In Node B, GSM module receives the cipher text transmitted from Node A. It gives cipher text to ARM board. The AES

Decryption code is dumped onto ARM board.AES decryption algorithm converts cipher text into original text. The plain text is given to computer and it is displayed.

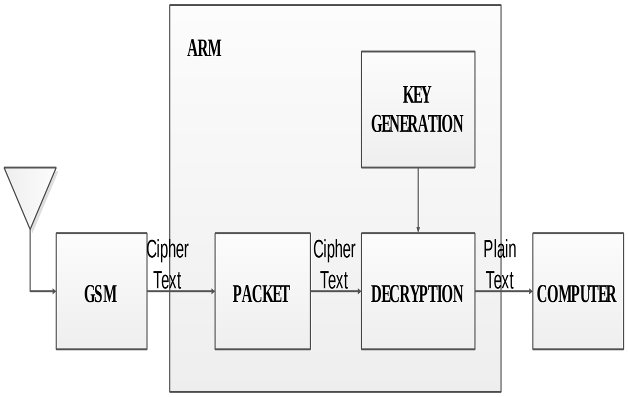


Fig. 4 Node B

A. Interfacing of ARM LPC 2148 board and Computer

The UART port of computer is connected to UART 0 port of ARM LPC2148 board. The AES encryption and decryption code is dumped onto ARM through UART port by using a flash burner called Philips flash utility V2.2.3.

B. Interfacing of ARM LPC 2148 board and GSM Module (SIM 300)

The UART 1 port of ARM LPC 2148 is connected to UART port of GSM Module. The GSM module is controlled by ARM through AT commands. The important AT commands are as follows:

- AT- It us used to check communication between GSM module and ARM
- ATE0-Command Echo
- AT+CMGF-This command is used to set the SMS mode. Either text or PDU mode can be selected by assigning 1 or 0 in the command
- AT+CMGS-This command sends a short message from the modem to the network
- AT+CMGR-Read message
- AT+CMGD-This command deletes a message from the location from SIM storage.

VI. RESULTS

The implementation of AES algorithm onto ARM for wireless network is successfully done. The TABLE II shows the time required for key expansion of 128 Bit AES for different rounds.

TABLE II
TIME REQUIRED FOR KEY EXPANSION OF 128 BIT AES

Rounds	Time (us)
10	24.92 usec
12	29.42 usec
14	33.92 usec

Fig 5. Represents the execution of each process in AES encryption. The total time required for AES encryption execution is 481.50usec.Here the time required for AES encryption execution is calculated for rounds 10. It shows execution time required for each process in %.

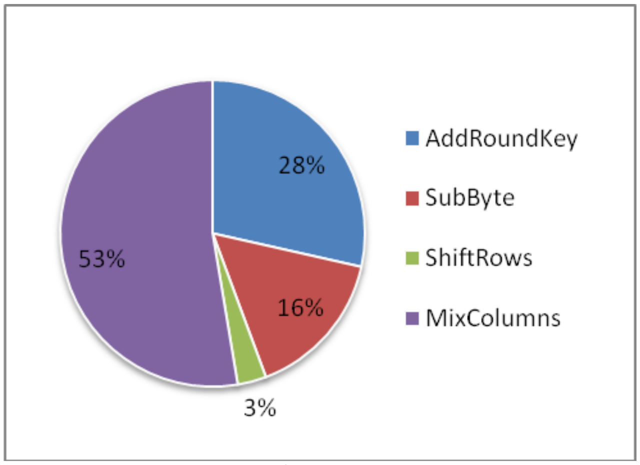


Fig. 5 Time required for each process in AES Encryption

Fig 6. Represents the execution of each process in AES decryption. The total time required for AES decryption execution is 1317.40usec. Here the time required for AES decryption execution is calculated for rounds 10.It shows execution time required for each process in %.

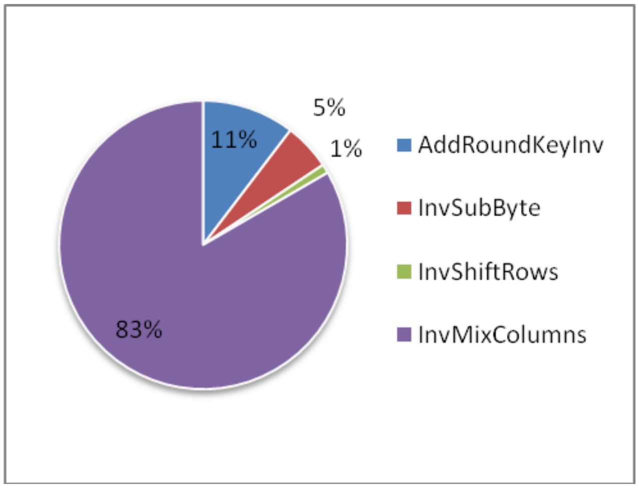
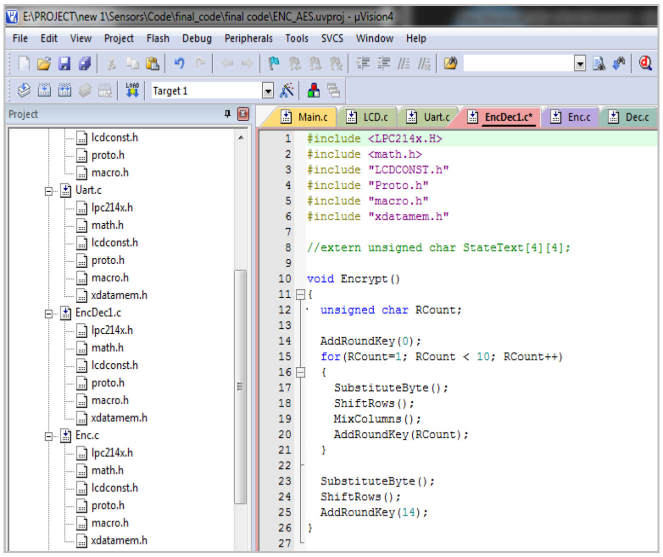


Fig. 6 Time required for each process in AES Decryption

The source code which is developed in embeddedC language the Fig 7 represents example for the source code of AES encryption and decryption algorithm.

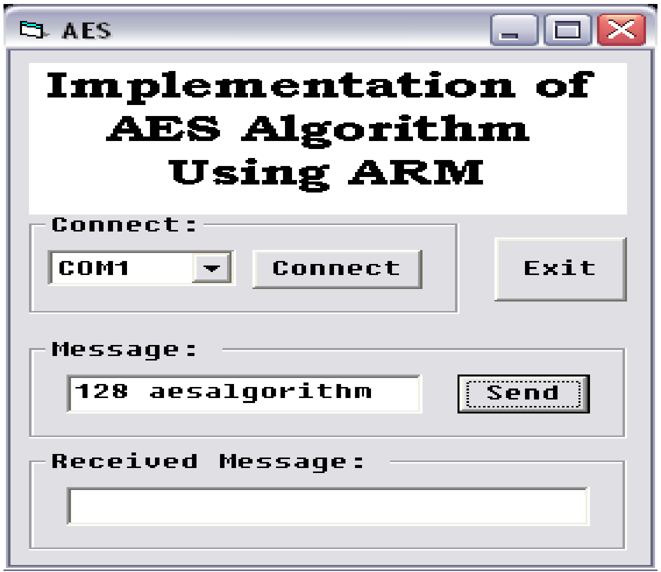
After developing the source code , burn the programming into the ARM processor by using the Philips flash utility V2.2.3.



```c
#include <LPC214x.H>
#include <math.h>
#include "LCDCONST.h"
#include "Proto.h"
#include "macro.h"
#include "xdatamem.h"

//extern unsigned char StateText[4][4];

void Encrypt()
{
  unsigned char RCount;

  AddRoundKey(0);
  for(RCount=1; RCount < 10; RCount++)
  {
    SubstituteByte();
    ShiftRows();
    MixColumns();
    AddRoundKey(RCount);
  }

  SubstituteByte();
  ShiftRows();
  AddRoundKey(14);
}

```

Fig. 7 AES Algorithm Code

We developed a GUI in visual basics. GUI contains Send message window and received message window. The message which is to be send (plain text) typed it into message send window. The Fig. 8 shows '128 aesalgorithm' is send.



Implementation of AES Algorithm Using ARM

Connect: COM1

Message: 128 aesalgorithm

Received Message:

Fig. 8 Plaintext before AES Encryption

In Fig. 9 second line of LCD shows the encrypted message (cipher text) and first line shows the encrypted message is send through GSM module.

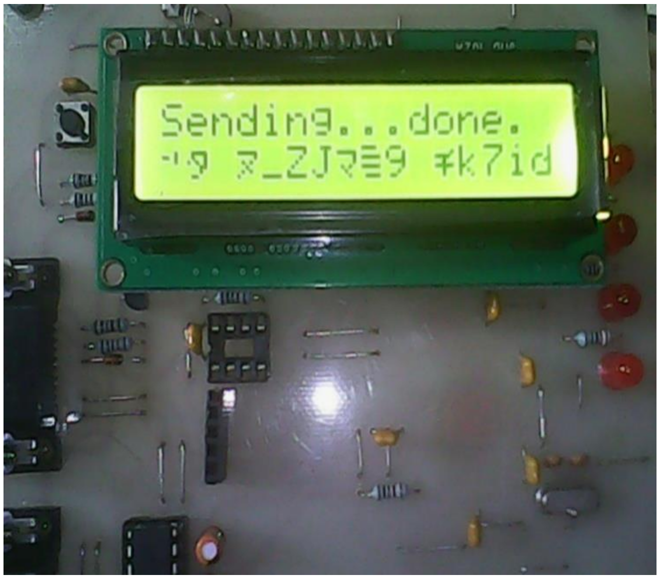


Fig. 9 AES Encrypted data

After receiving the encrypted message it is converted into plaintext by AES Decryption. In Fig. 10 second line of LCD shows original plain text which is transmitted. The first line of LCD shows the plaintext is further sending to computer.

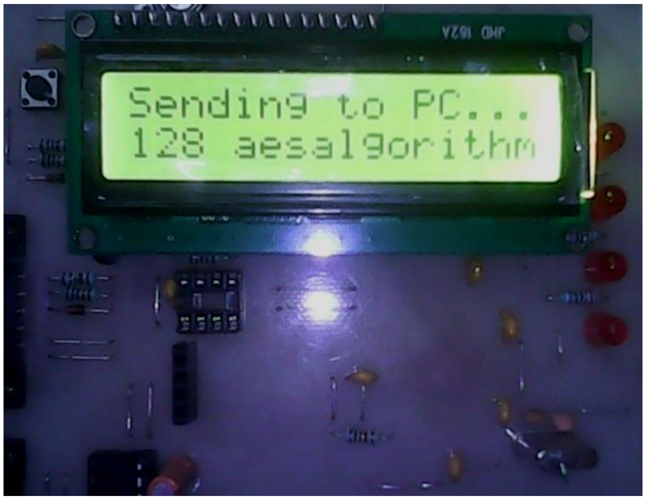
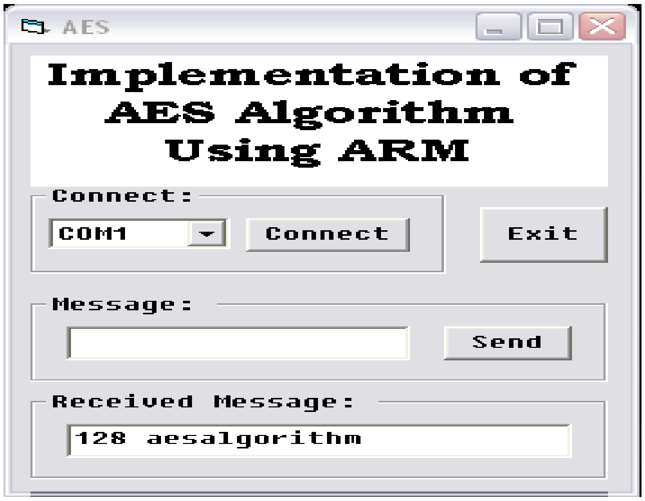


Fig. 10 AES Decrypted data

In Fig. 11, In Received message window the original plain text is received by AES decryption.



Implementation of AES Algorithm Using ARM

Connect: COM1

Message:

Received Message: 128 aesalgorithm

Fig.11 Plaintext after AES decryption

VII. CONCLUSION

In this paper, we have implemented the AES encryption and decryption algorithm on to ARM LPC2148.The time required for key expansion of 128 bit AES is calculated for different rounds. The execution time for AES encryption and decryption is calculated for rounds 10.

After AES encryption the data is send from wireless Node A and it received on wireless Node B. The AES provides security data confidentiality, integrity and authentication. Thus the communication between both wireless node is secured.

REFERENCES

[1]. N. Sloss, D. Symes, and C. Wright, ARM System Developer's Guide, Designing and Optimizing System Software, Morgan Kaufmann, 2004.

[2]. Journal of research of the NIST, volume 106, November 3, May- June 2001

[3]. NIST, Advanced Encryption Standard (AES), (FIP PUB 197), November 26, 2001.

[4]. J. Daemen and V. Rijmen, AES Proposal: Rijndael (Version 2). NIST AES

[5]. D. Popescu, R. Dobrescu, M. Nicolae, and A. Iordan, "Communication Infrastructure Selection Criteria for Alerting Systems Implemented through Mobile Sensor Networks", Proceedings of the 10th WSEAS International Conference on AUTOMATION & INFORMATION ICAI 09, Prague, March 23-25, 2009, pp. 354-359.

[6]. M. McLoone, J. McCanny, "High Performance Single-Chip FPGA Rijndael Algorithm Implementations," Proceedings Cryptographic Hardware and Embedded Systems Workshop, CHES, Paris, May 2001.

[7]. R. Ashruf et al, Reconfigurable Implementation for the AES Algorithm, Delft University of Technology, Netherlands, 2005.

[8]. G. Bertoni, L. Breveglieri, P. Fragneto, M. Macchetti and S. Marchesin, "Efficient Software Implementation of AES on 32-bit Platforms," CHES 2002, LNCS 2523, pp. 159–171, 2003.

[9]. K. Atasu et al, Efficient AES Implementation for ARM Based Platforms, ACM, 2004

[10]. B. Gladman, A specification for Rijndael, the AES Algorithm. Available at http://fp.gladman.plus.com, May 2002.

[11]. T.Ravichandra Babu, K.V.V.S.Murthy, G.Sunil , "AES Algorithm Implementation using ARM Processor",2nd International Conference and workshop on Emerging Trends in Technology (ICWET) 2011.

[12]. W. Stallings, Cryptography and network security.

[13]. http://www.esat.kuleuven.ac.be/~rijmen/rijndael

BIOGRAPHY

Mr. Vinayak Bajirao Patil did his B.E (Electronics) from Shivaji University, Kolhapur in the year 2010. He is pursuing her M-TECH (Electronics) from Department of Technology, Shivaji University, Kolhapur. He has a total 02 years of experience in teaching. He has presented 1 paper in National Conference. He has attended number of workshops on various subjects.

Prof. Dr. U.L Bombale Has received PhD from Dhirubhai Ambani Institute of Information & Communication Technology, (DA-IICT) Gandhinagar, Gujarat, India, under the guidance of Dr. Sanjeev gupta. M.E in Electronics & Telecommunication in 1994 from COE, Pune, and currently he is working as a Professor in Dept. of Technology, Shivaji university, Kolhapur (India).