IMPLIMENTATION OF OPTIMUM ENCRYPTION ALGORITHM USING CHAOTIC SYSTEMS WITH THE APPLICATION OF PRIMITIVE ROOT THEOREM

Jatin M. Patel [1] Anuj M. Patel2
[1]Electronics & Communication Department
[2]Gujarat Technological University, [1]SSESGI, Rajpur , Kadi, India.

Abstract—This paper describes the secure wireless communication in detail. This process is very useful in modern applications of wireless communication. First of all, we take the binary data form computer (or internet). This data is encrypted using encryption key and then it is transmitted using radio frequency transmitter. At the opposite side receiver receives the data and receiver decrypt the data using same encryption key. In this project we have added additional security by changing the encryption key every time. In this project I have added some features like it can transmit live image, videos at the speed of Bluetooth.

Keywords: AES Algorithm, Microcontroller, cipher key, encryption function, decryption function.

I. INTRODUCTION

This paper is about the secure wireless communication over RF. The main advantage of this paper is that the data cannot be received until and unless you don't have receiver code that is compatible to transmitter. At the transmitter, computer will be attached to microcontroller which is used to input the data (image, video or text file).This data is encrypted and then it is transmitted using radio frequency transmitter. The data which will be entered at the transmitter will also be displayed on LCD for convenient entry. At the receiving end when the encrypted data is received then this message is decrypted by the microcontroller and is displayed on the LCD. The mode of communication that I have used in my project is radio frequency channel[We have used two schemes of data encryption and decryption. One is the Monoalphabetic scheme and the other is Polyalphabetic scheme. The RF models that we have used for data transmission and reception works at 433MHz]There is one additional security added to this algorithm, and that is Encryption key will be changed at every time of sending the data from transmitter to receiver. But here we are transmitting the encryption key through wireless. In effect of transmitting encryption key everyone can receive that key and misuse it. So we are dividing the encryption key in different parts and transmit at different frequency. So authorized receiver can only receive the encryption key and use it to decrypt the data. This will give more security to transmitting the data. And it is also doing one type of encryption to encryption key.

II. AES ALGORITHM

Cryptography plays an important role in the security of data. It enables us to store sensitive information or transmit it across insecure networks so that unauthorized persons cannot read it. The basic unit for processing in the AES algorithm is a byte (a sequence of eight bits), so the input bit sequence is first transformed into byte sequence. In the next step a two-dimensional array of bytes (called the State) is built. The State array consists of four rows of bytes, each containing Nb bytes, where Nb is the block size divided by 32 (number of words). All internal operations (Cipher and Inverse Cipher) of the AES algorithms are then performed on the State array, after which its final value is copied to the output (State array is transformed back to the bit sequence). The input and output for the AES algorithm each consist of sequences of 128 bits (digits with values of 0 or 1). These sequences will sometimes be referred to as blocks and the number of bits they contain will be referred to as their length. The Cipher Key for the AES algorithm is a sequence of 128, 192 or 256 bits. The AES algorithm consists of ten rounds of encryption, as can be seen in Figure 3 First the 128-bit key is expanded into eleven so-called round keys, each of them 128 bits in size. Each round includes a transformation using the corresponding cipher key to ensure the security of the encryption.

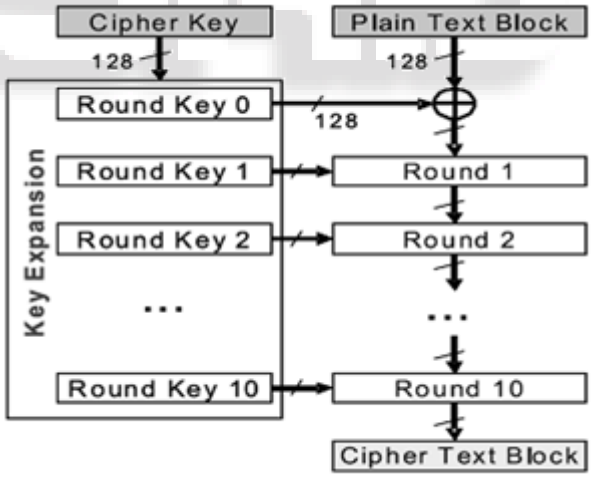


Fig. 1: AES Algorithm structure

After an initial round, during which the first round key is XORed to the plain text (Addroundkey operation), nine equally structured rounds follow. Each round consists of the following operations:

- Substitute bytes
- Shift rows
- Mix columns
- Add round key

The tenth round is similar to rounds one to nine, but the Mix columns step is omitted.

A. Structure of Key and Input Data

Both the key and the input data (also referred to as the state) are structured in a 4x4 matrix of bytes. Figure 2 shows how the 128-bit key and input data are distributed into the byte matrices.

B. Substitute Bytes (Sub bytes Operation)

There are different ways of interpreting the Sub bytes Operation. In this application report, it is sufficient to consider the Sub bytes step as a lookup in a table. With the help of this lookup table, the 16 bytes of the state (the input data) are substituted by the corresponding values found in the table (see Figure).

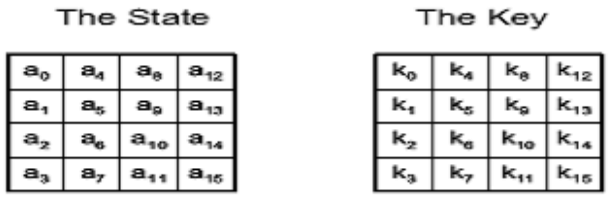


Fig. 2: Structure of the key and the state

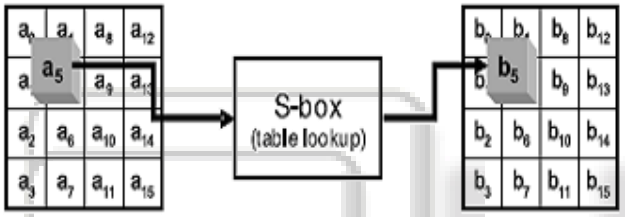


Figure: 3 Sub byte Operation

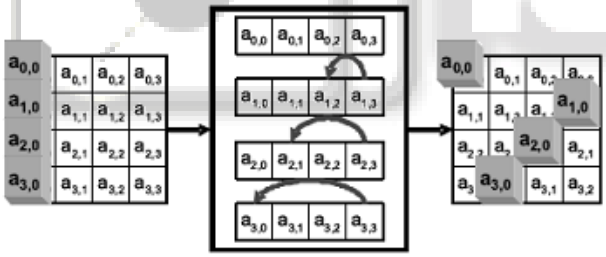


Figure: 4 Shift rows Operation

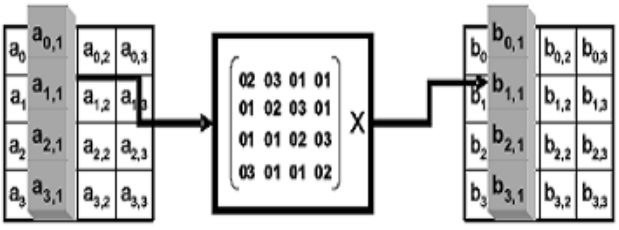


Figure:5 Mix columns Operation

C. Shift Rows (Shift rows Operation)

As implied by its name, the Shift rows operation processes different rows. A simple rotate with a different rotate width is performed. The second row of the 4x4 byte input data (the state) is shifted one byte position to the left in the matrix, the third row is shifted two byte positions to the left, and the fourth row is shifted three byte positions to the left. The first row is not changed.

D. Mix Columns (Mix columns Operation)

Opposed to the Shift rows operation, which works on rows in the 4x4 state matrixes, the Mix columns operation processes columns. Transformation in the Cipher that takes all of the columns of the State and mixes their data (independently of one another) to produce new columns shown in figure 5.

E. Add Round Key (Addround key Operation)

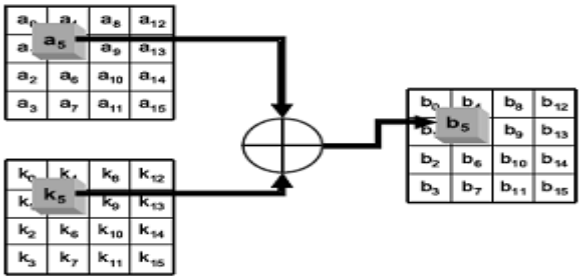


Figure: 6 Addround Key Operation

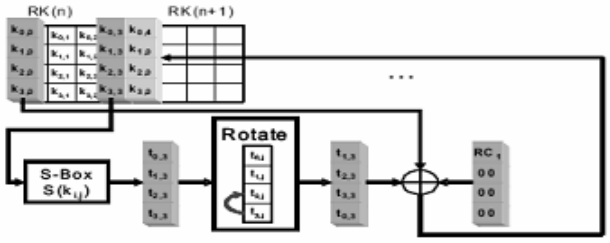


Figure: 7 Expanding First Column of Next Round Key

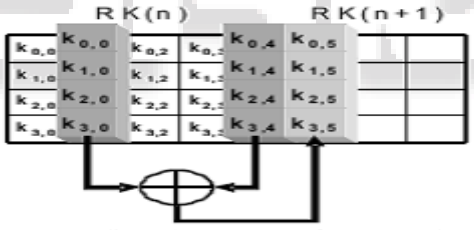


Figure: 8 Expanding Other Columns of Next Round Key

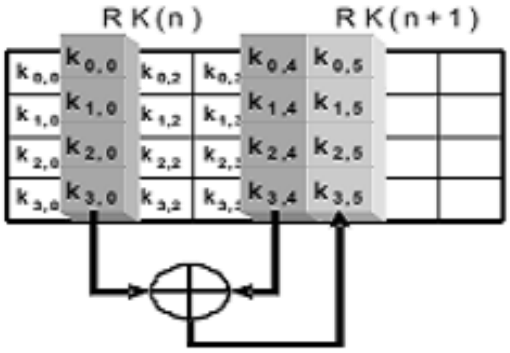
The Addroundkey operation is simple. The corresponding bytes of the input data and the expanded key are XORed (see Figure 6).As previously mentioned, Key expansion refers to the process in which the 128 bits of the original key are expanded into eleven 128-bit round keys

(a). To compute round key (n+1) from round key (n) these steps are performed: Compute the new first column of the next round key as shown in Figure 7:First all the bytes of the old fourth column have to be substituted using the Sub bytes operation. These four bytes are shifted vertically by one byte position and then XORed to the old first column. The result of these operations is the new first column.

(b). Columns 2 to 4 of the new round key are calculated as shown:

– new second column] = [new first column] XOR [old second column]

- [new third column] = [new second column] XOR [old third column]
- [new fourth column] = [new third column] XOR [old fourth column



. Figure: 9 Expanding Other Columns of Next Round Key

F. AES Encryption Flow Chart

In the flow chart of AES encryption algorithm round counter for 128 bit is 10. Then all other operation like key addition, S-Table Substitution, Encode Row Shift, Encode Mix Column are performed.

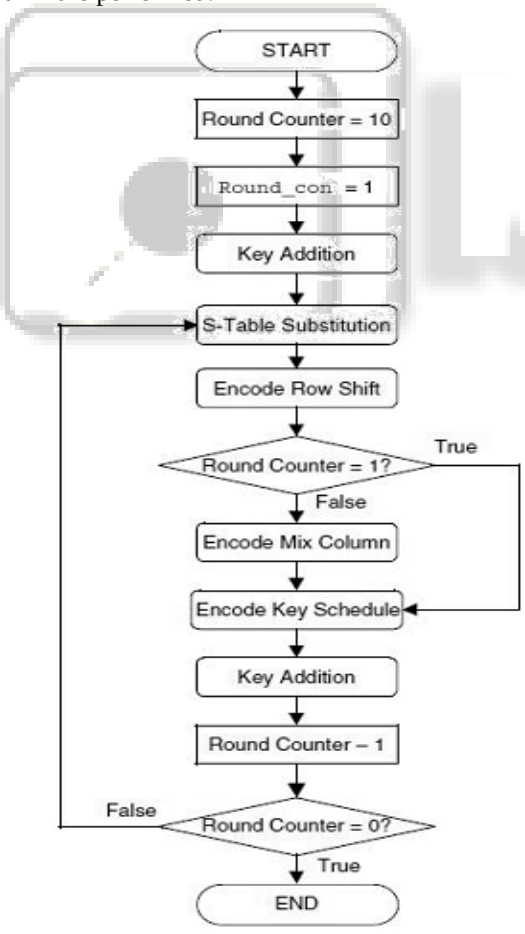


Figure: 10 AES Encryption Flow Chart

III. SYSTEM DESIGN

Data is transmitted from transmitter to receiver and vice versa when data is given through the keyboard scan code of keyboard is used. If data is given through the PC then prolific USB to UART IC pl2303 is used. Given data is load in At Mega 16 using emulator. Then data is transmitted from transmitter to receiver through the wire antenna. Prolific pl2303 is used for Communication interface between USART based serial port of microcontroller and USB port of computer. Visual basic based GUI makes it easy to pre-store the response of transmitter user.

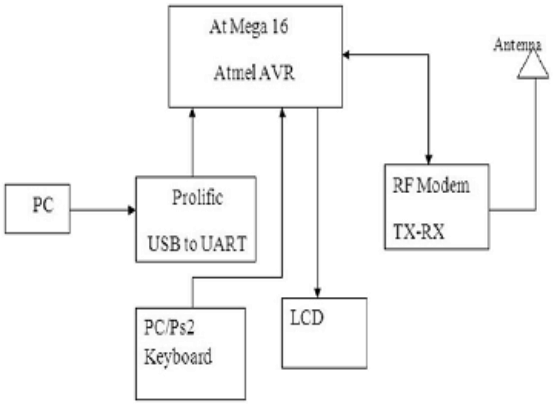


Fig.: 11 Block Diagram of system

RFM12 is a low cost FSK transceiver IC which integrate all RF function in single chip. It only need a MCU, a crystal, a decoupled capacitor and antenna to build a high reliable FSK transceiver system. The operation frequency can cover 300 to 1000 MHz RF12 supports a command interface to setup frequency , deviation, o/p power and also data rate. No need any hardware adjustment when using in frequency hopping application.

The PL-2303 operates as a bridge between one USB port and one standard RS232 Serial port. This device is also compliant with USB power management and remote wakeup scheme. Only minimum power is consumed from the host during suspend. By integrating all the functions into the SSOP-28 package, this chip is suitable for cable embedding. Users just simply hook the cable into PC or hub's USB port, and then they can connect to any RS-232 devices.

IV. SYSTEM FLOWCHART

The input data is given through the PC. For that Visual Basic Graphical interface is used. When data is given through the PC USB to UART converter IC pl2303 is used. Then data is loaded in At Mega16.

For the security of data cryptography algorithm is used. After that encrypted data is displayed on LCD. When data is given through the keyboard scan code is used. For the data communication we worked on 433 MHz frequency.

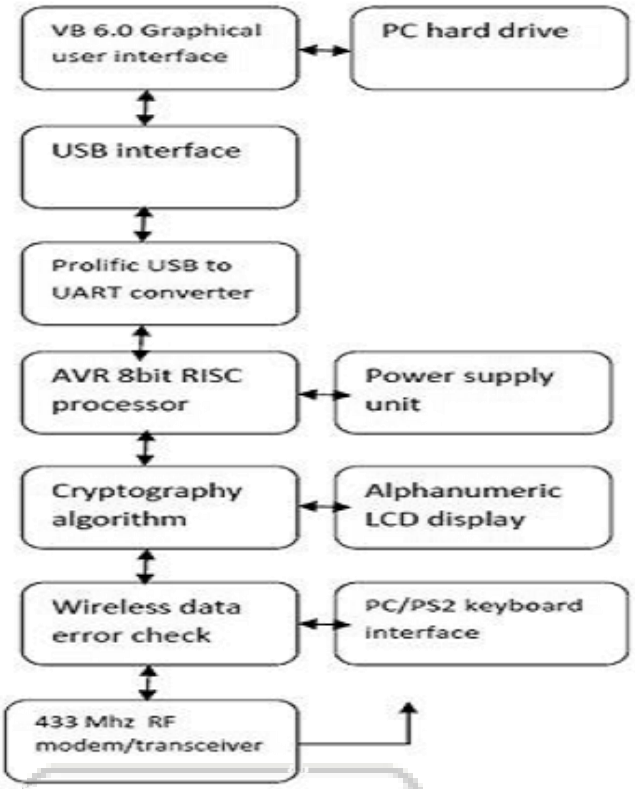


Figure: 12 Flow chart of system

V. SOFTWARE FLOWCHART

This software will be implemented in Embedded C. In this software algorithm there are three important functions.

(a). Control Function
(b). Encryption Function
(c). Decryption Function

Control function contains user interface algorithm. In the control function there are two programs. One is visual basic based software which can be installing in host computer. And another is implemented in embedded c. VB (visual basic) software will get the inputs and commands from user and transfer to the embedded c software. It will do operation on inputs. It will basically take the decision whether controller has to call encryption program or decryption program. And VB software also gets the input from embedded c software (microcontroller). And display to the user. Embedded c software will be installed in microcontroller.

Encryption function contains encryption algorithm. It will generate the encryption key. And using this encryption key it will encrypt the input data. And then it will give this encrypted data and encryption key to control function.

Control function divide the encryption key in three parts and send this every part at different frequency. At the opposite side receiver (Decryption function) has knowledge about this frequency range and it will receive different three parts and combine this three parts it will get the encryption key. After this process transmitter starts to transmit the encrypted data.

Decryption function contains decryption algorithm. After receiving encryption key it will decrypt the data using this encryption key. After this process it will pass decrypted data to control function.

VI. SIMULATION RESULT

Simulation of Encryption code in µcontroller

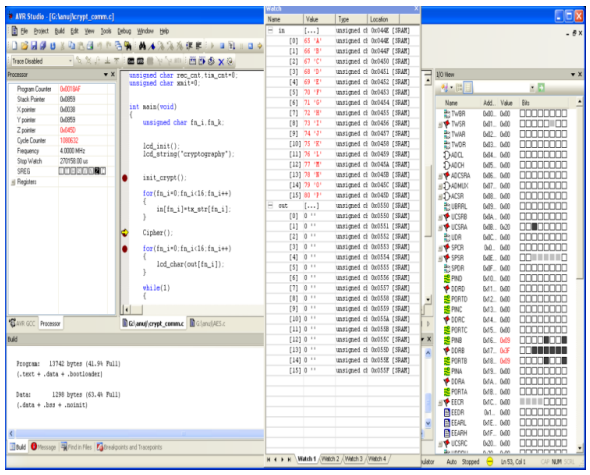


Fig. 13: Simulation of Encryption code in µcontroller

Simulation of Decryption code in microcontroller.

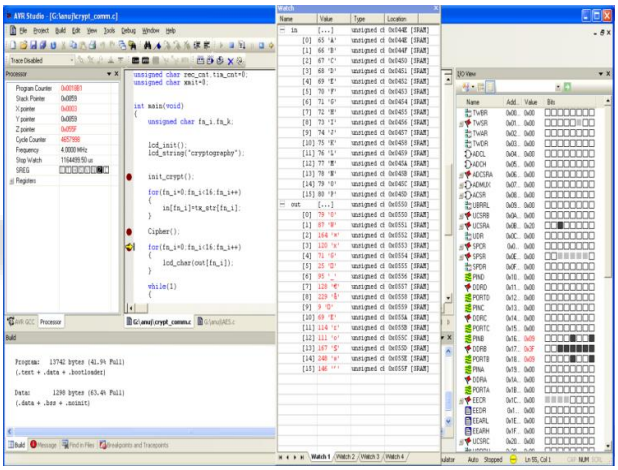


Fig. 14: Simulation of Decryption code in microcontroller.

SIMULATION IN TURBO C

Encryption function

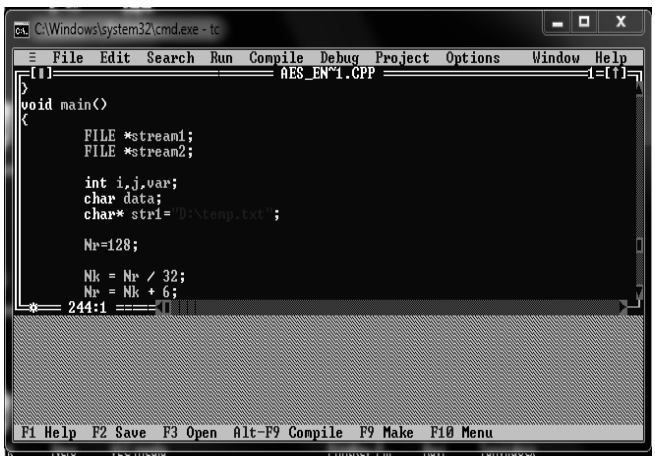


Fig. 15 : Encryption function

Source file

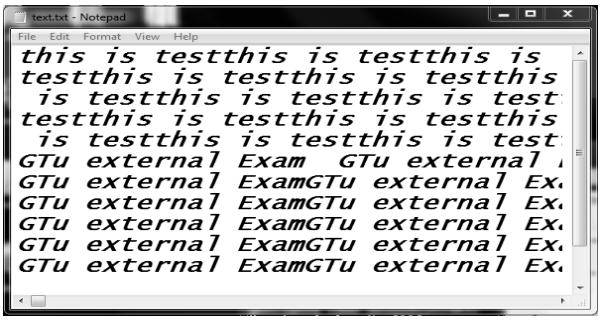


Fig. 16: Source file

Encrypted file

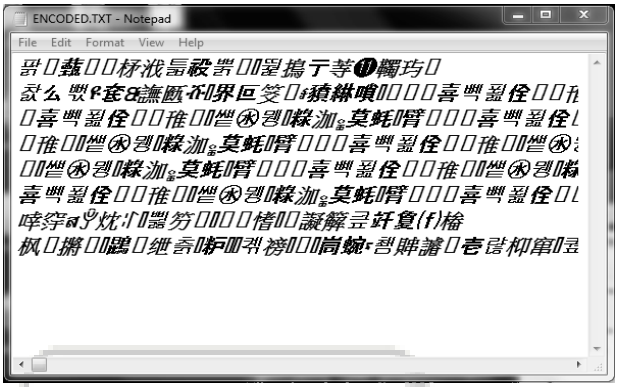


Fig. 17: Encrypted file

Decryption function

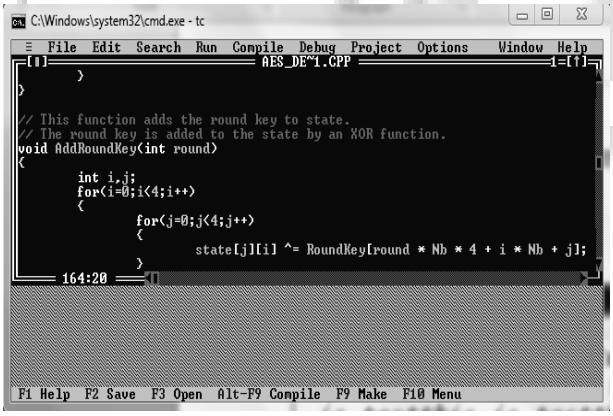


Fig. 19: Decryption function

Decrypted file

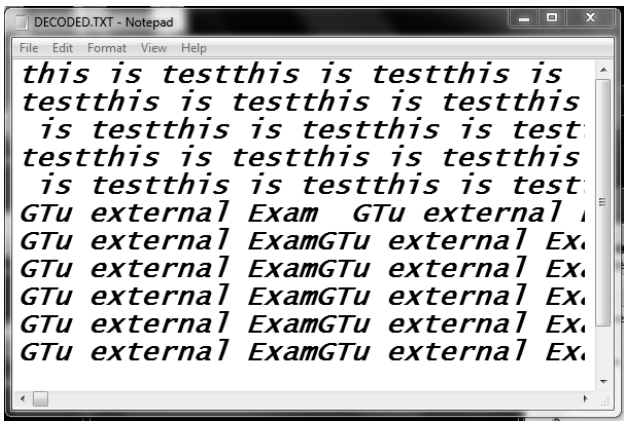


Fig. 20: Decrypted file

VII. CONCLUSION

After so much work out on this project, we finally got our devices work in the way that we specified. It is able to send or receive messages and handle all handshaking problems and errors generated. The data is sent from the transmitter after encryption and at the receiving device the data is decrypted according to the specific algorithm. If we were given this project again, we will definitely search the transmitting and receiving modules which operates at different frequencies. This would simplify our logic and the state transition of the project because we will get rid of handshaking problems and errors. We will also make a change so that our device can communicate with multiple devices. This should not be too difficult but we have to program the target's identification number. In this way only our desired target will only pick up the transmission.

REFERENCES

[1] M. Haleem, C. Chetan, R. Chandramouli and K.P. Subbalakshmi,Opportunistic Encryption: A Trade-Off between Security and Throughput in Wireless Networks," IEEE Transaction on Dependaple and Secure Computing, vol. 4, no. 4, pp.313-324, Oct 2007.

[2] Schneier, "Applied Cryptography," China Machine Press, Second Edition, 2007, pp. 188-210.

[3] W. M. Daley and R. G. Kammer, "Data encryption standard (DES)," U.S.DEPARTMENT OF COMMERCE/National Institute of Standards andTechnology, Oct. 25, 1999.

[4] 2009 International Forum on Information Technology and Applications A New Embedded Encryption Algorithm for Wireless Sensor Networks.

[5] http://en.wikipedia.org/wiki/Triple_DES, Wikipedia, the free encyclopedia.

