Microcontroller Based Cryptosystem with Key Generation Unit using TEA Algorithm

Shilpa G, Siva S. Yellampalli

Abstract— This work focuses on the light weight security algorithm Tiny Encryption in order to increase the speed and to reduce the hardware complexity. Algorithm TEA which can be implemented in microcontroller to adapt with many real time constraints such as memory, data loss and low cost. The additive feature of this proposed system is that it uses Key Generation Unit (KGU) to produce the random key to make it optimal for sensitive data transfer in many real-time applications. This above work uses microcontroller and the performances of this cryptosystem is analyzed by implementing the cryptographic algorithm TEA with key generation unit. The work extends with implementation of serial (UART) as mode of communication to transfer the data from encryption unit to decryption unit.

Index Terms—TEA, UART.

I. INTRODUCTION

Today's computing systems, which consist of a broad range of processors, communication networks and information repositories, are vital to the operation of many sectors of our society, from financial and manufacturing to education and healthcare. The explosive growth and the open nature of the Internet and e-commerce have caused organizations to become more vulnerable to malicious electronic attacks than ever before. With the increasing quantity and sophistication of attacks on IT assets, companies have been suffering from breach of data, loss of customer confidence and job productivity degradation, all of which eventually lead to loss of revenue. Many network security measures have been proposed to counter those attacks in order to guarantee the confidentiality, integrity, authenticity and availability of resources. Data Security is a challenging issue of data communications today that touches many areas including secure communication channel, strong data encryption technique and trusted third party to maintain the database. The rapid development in information technology, the secure transmission of confidential data herewith gets a great deal of attention. The conventional methods of encryption can only maintain the data security. The information could be accessed by the unauthorized user for malicious purpose. Therefore, it is necessary to apply effective encryption/decryption methods to enhance data security. Cryptography is a part of information security. It is an art of securing the data. It is mainly concerned with storing and transmitting the information safely over the insecure medium like Internet by encoding text data into a form non recognizable format with the help of various encryption algorithms and only the intended user will be able to convert it into original text.

Revised Version Manuscript Received on June 19, 2015.

Shilpa G, Final Semester Student, M.Tech. – VLSI Design and Embedded Systems, Department of PG Studies, VTU Extension Centre, UTL Technologies Ltd. Bangalore, India.

Dr. Siva S. Yellampalli, Department of PG Studies, VTU Extension Centre, UTL Technologies Ltd. Bangalore, India.

The process which converts original data into the unreadable form is called encryption process as shown in figure 1.

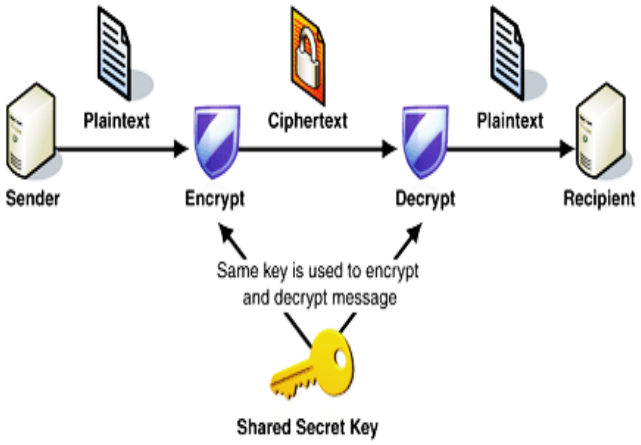

Figure 1. Encryption and Decryption

In the existing system, the hardware implementation of block ciphers has limited feasibility in scheduling the key which is the primary resource for high secured data transfer. Since the existing system uses predefined key for the encrypting process, the system can offer narrowed security level though they use complex security algorithms. The major drawback in the existing system is that there is no key generation unit to increase the efficient change of key parameter for a secured data transfer. The proposed system implements the above statements using the light-weighted, secure and efficient block cipher Tiny Encryption Algorithm (TEA).

II. RELATED WORK

Cryptography is the practice and study of techniques for secure communication in the presence of third parties (called adversaries) [1]. Tiny Encryption Algorithm was designed David Wheeler and Roger Needham of the Cambridge University Computer Laboratory and presented in November 1994. It is designed to encrypt data through many iteration cycles rather than memory consuming arrays of data. TEA [2] is a short algorithm which will run on most machines and encipher safely. The performance of these block ciphers are compared with that of the Advanced Encryption Standard (AES) implementation and it has been proved that TEA consumes less memory than all the others [3]. There are some goals of cryptography that are given below [4]:

1) Authentication: Sender and data receiver must be authenticated before sending and receiving data.

2) Confidentiality: The user who is authenticated, can access the messages

3) Integrity: Data is free from any kind of modification between sender and receiver.
4) Non-Repudiation: The sender the receiver cannot deny that they had sent a message.
5) Service Reliability: Attackers can attack on secure systems, which may affect the service of the user.
The data encipherment only cannot provide security to any system. To decrypt the enciphered information the key has to be shared amongst all the legitimate users. The communication involves two or more participants. Hence the importance of secret sharing and management of the keys is a vital problem. Large numbers of methods have been proposed in the literature to solve this problem. These methods of secret sharing use the cryptographic protocols[5]. In this work, serial communication is used for communication to store the generated keys in EEPROM. UART communication is used for transmitting the encrypted key from encryption unit to decryption unit. The Universal Asynchronous Receiver/Transmitter (UART) controller is the key component of the serial communications subsystem of a computer. The UART takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes [8].

III. PROPOSED SYSTEM

The block diagram of the proposed system is shown in figure 2. It has 3 main units:
1. Encryption Unit
2. Decryption Unit
3. Key Generation Unit

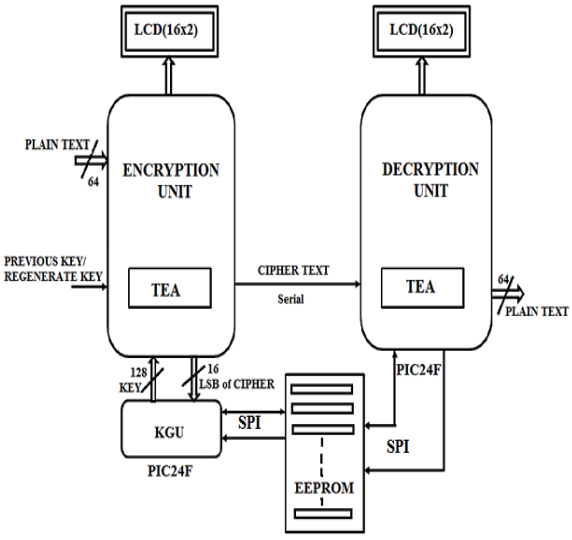

Figure 2. Block diagram of proposed system

1. Encryption unit:

The encryption unit consists of a development board with microcontroller, LCD, EEPROM, Keypad and RS232 to UART circuit. The microcontroller used in this work is PIC24FJ128GA204.The microcontroller stores the routines for encryption, LCD display functions, EEPROM routines to store the key and UART routines for transmission of data from encryption unit to decryption unit. The figure 3 shows flow chart for the encryption unit. The encryption unit accepts input data from the keypad interfaced with the encryption unit. The encryption can be carried out using the predefined random key or the new key can be generated using the key generation unit. The data will then be encrypted and the encrypted data will be displayed on LCD. Once the encryption is done, the encrypted data will be sent to the decryption unit.

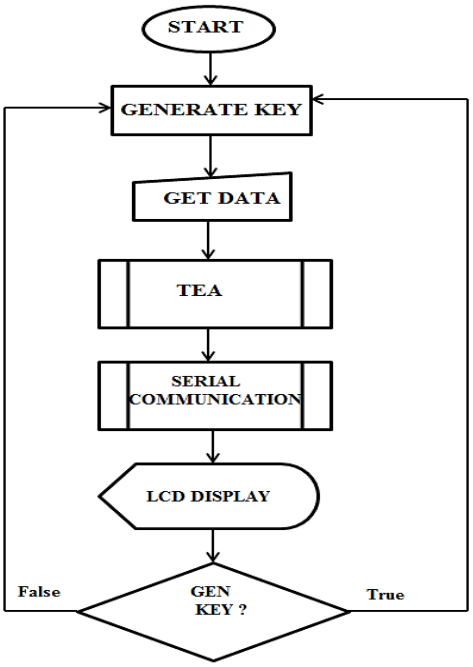

Figure 3. Flowchart for Encryption Unit

2. Decryption Unit:

Decryption Unit receives the encrypted data from the encryption unit and decrypts the data using the key from EEPROM and then displays the original data on LCD.
The flowchart for Decryption Unit is as shown in Figure 4.

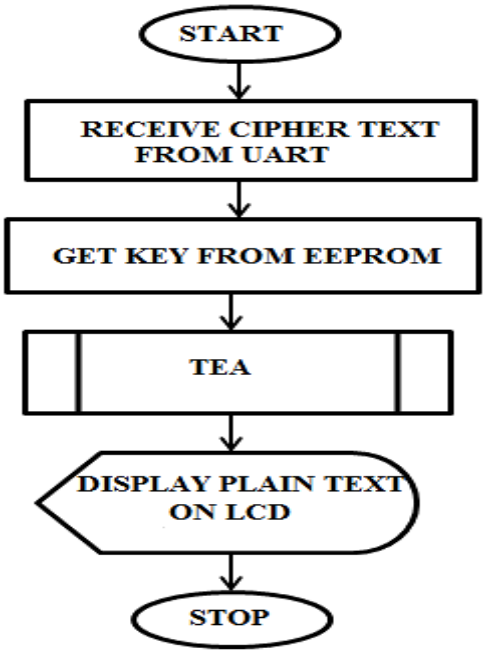

Figure 4. Flowchart for Decryption Unit

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

3. Key Generation Unit:

The KGU is implemented using timers in the microcontroller to generate the random bits. The generated random numbers acts as a key for block cipher and is stored in EEPROM, so that the key is secured and it is then transferred to the decryption unit. The KGU uses the in-built timers (T0 & T1) which accept the input from the crystal oscillator by dividing its value by12. In order to generate 128 bit key from 16 bit timers, it has to take 8 samples from the timer.

IV. TEA

The Tiny Encryption Algorithm (TEA) block cipher was designed with speed and simplicity in mind. It is a variant of the Feistel Cipher. TEA operates on a 64 bit block of data that is then split up into two 32 bit unsigned integers during the encryption process. TEA uses a 128 bit key, and a magic constant is also utilized which is defined as 2^32(the golden ratio). This quantity looks like 2654435769 when expressed as an integer. The original cipher spec was written by Roger Needham and David Wheeler. TEA operates on 64 (block size) data bits at a time using a 128-bit key with 32 rounds. TEA is an iteration cipher, where each round i has inputs M0[i-1] and M1[i-1] as in (2) and (3), which is derived from the previous round. The subkey K[i] is derived from the 128 bit overall K. The constant delta (_) in (1), is the derivative of golden number ratio to ensure that the subkeys are distinct.

$$\partial = (\sqrt{5} - 1) * 2\text{^}31 = 9E3779B9h \quad - (1)$$

$$M0[i] = M0[i-1] \boxplus F(M1[i-1], K[0, 1], DELTA[i])$$

$$M1[i] = M1[i-1] \boxplus F(M0[i-1], K[2, 3], DELTA[i])$$

A. Encryption Routine

The figure 6 shows the architecture for TEA encryption process.

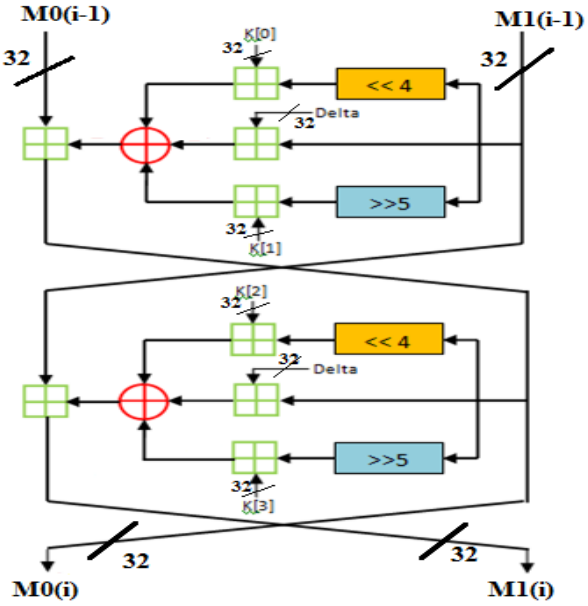

Figure 6. TEA Encryption Process

The diagram above shows 2 Feistel rounds of TEA. These two Feistel rounds make up one cycle of TEA. The cipher starts with a 64 bit data block that is split up into two 32 bit blocks. TEA has a 128 bit key that is split up into four 32 bit subkeys, which can be seen as K[0-3]in the diagram.

B. Decryption Routine

Similar operations are performed for decryption process which is described in figure 7. In this case, the constant delta value DELTA(i-1) is "C6EF3720", where „i" represents the number of iterations. In each iteration, the delta value "9E3779B9" is subtracted with the constant delta value.

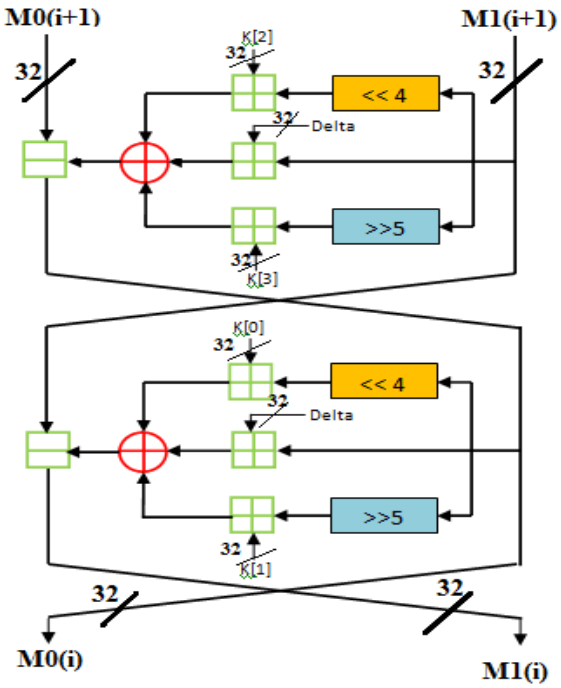

Figure 7. TEA Decryption Process

Since, the TEA has a Feistel structure the reverse operation of encryption process is performed to obtain the plain text in the decryption process.

V. RESULTS

A. Experimental setup

The experimental setup consists of 2 Explorer 16 boards with 2 microcontrollers, one for Encryption Unit and the other for Decryption Unit connected together using a UART cable for transmission of encrypted data from Encryption Unit to Decryption Unit. Figure 8 gives the actual implementation of experimental setup for hardware testing.

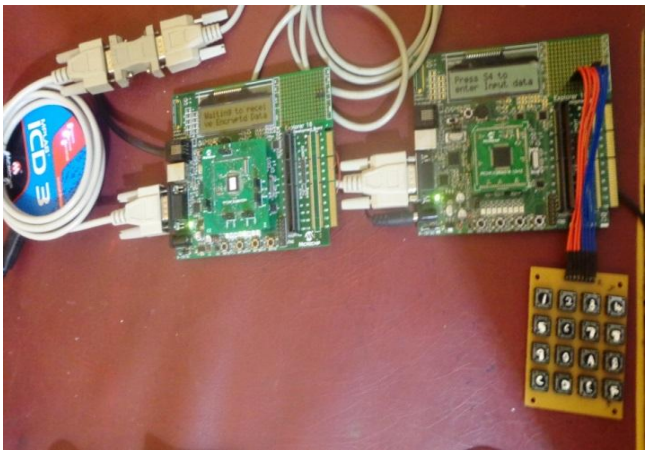

Figure 8. Experimental Setup for Hardware Testing3

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

B. Hardware testing results

The setup for hardware testing consists of two units, encryption and decryption unit. Encryption unit consists of explorer 16 board interfaced with keypad for entering the input data and LCD for displaying the input data and encrypted data. The decryption unit also consists of Explorer 16 board interfaced with LCD to display the decrypted or original data. After powering up the encryption unit, input data should be entered. After that, an option for either using the stored key or generated key needs to be selected. If the stored key option is selected, the stored key will be used for encrypting the input data and the encrypted data will be displayed on LCD. If the key generation option is selected, 8 keys will be generated and u can select to use one of the 8 generated keys. The selected key out of eight generated random keys will be used for encryption and the encrypted data will be displayed on LCD. To verify if the encrypted data is correct, it can be checked using online TEA encryption tool or it can be passed to the decryption unit for decryption. To check decryption unit, switch S2 is pressed and the data will be transferred to the Decryption unit through UART. The encrypted data will be decrypted and it will be displayed on LCD. It will be compared with the input data and the working of TEA encryption algorithm can be verified. By selecting the different keys for encryption from the key generation unit, the random encrypted results can be obtained and it shows that TEA encryption alone with key generation unit is more secure form of encryption. Figure 9 and 10 represents the output of encryption and decryption data on Explorer 16 board. *Note: 3.* Colour printing required

Figure 9. Encryption Output on Explorer 16 Board3

Figure 10. Decryption Output on Explorer 16 Board3

C. Simulation results

MPLABX IDE is used for the implementation of TEA encryption and decryption algorithms and it is simulated using the MPLABX simulator and the results of TEA encryption and decryption units are shown respectively in figures 11 and 12.

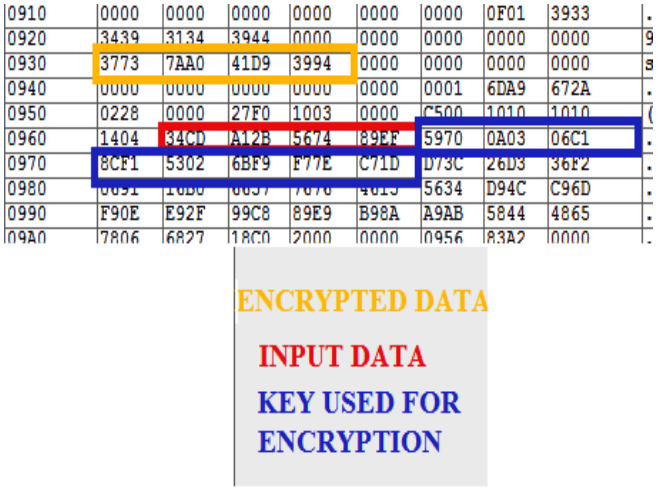

Figure 11. Simulation results of TEA encryption unit showing Input data, Key used for encryption and Encrypted data

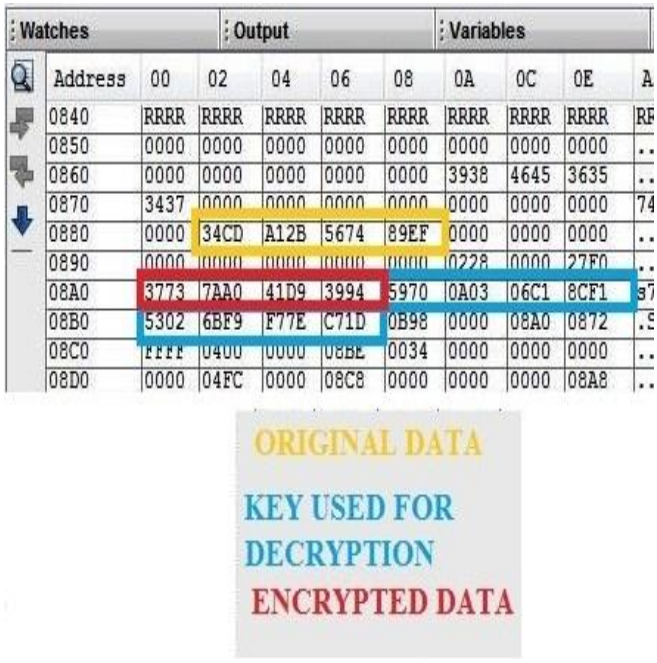

Figure 12. Simulation results of TEA encryption unit showing Encrypted data, Key used for decryption and Decrypted data

D. Result of the Key Generation Unit

The Key generation unit is implemented using timers to generate random key to be used for encryption. At a time, key generation unit generates 8 different keys. Control unit is used to select one of eight random keys to be used for encryption. Sample of 8 keys generated at a time is as shown table below:

VI. CONCLUSIONS

The system employs the utilization of TEA algorithm, which is a light-weight block cipher and offers low power and area consumption with moderate security. The additional feature of this system is the usage of KGU to generate the random key which still improves the security of data transfer. Therefore, the system is suitable to implement in the high secured low cost applications. The implementation of TEA using PIC microcontroller is to reduce the cost of the system. The application includes wired communication for secure data transfer. As indicated by the very low line of code counts, the algorithm is indeed simple to implement in varied languages. The porting of TEA to other languages is simple and straightforward. No specialized API functions are required that might need implementing before use. Any language that supports bitwise operations will execute the TEA algorithm with ease. When code size is extremely critical, TEA seem to be a reasonable choice. Testing the random numbers is very important in all cryptographic applications. The key generation unit is tested by generating a large number of keys and using those keys for data encryption. The strength of TEA emerges quite significantly. TEA seems to be highly resistant to differential cryptanalysis (Biham et al., 1992) and achieves complete diffusion (where a one bit difference in the plaintext will cause approximately 32 bit differences in the cipher text). Time performance on a workstation is very impressive. The data size for TEA encryption is 64-bits and the key size is 128 bits. The encryption and decryption cycles/block is 6200.The code size required for TEA is around 1140 bytes and it is 40.3% of AES encryption. The throughput of TEA at 4MHz is 40.8 and it is 53% of AES.

FUTURE WORK

For communication of data from encryption unit to decryption unit, wired communications can be extended to include wireless transmission via RF signals or Bluetooth. The Block TEA and XXTEA variants are recommend for encrypting large amounts of data. Many other techniques like random number generation using noise can be used to generate keys for encryption to improve security of the data transmission.

REFERENCES

1. Dr. Deepali Virmani, Nidhi Beniwal, Gargi Mandal, Saloni Talwar, "Enhanced Tiny Encryption Algorithm with Embedding", BPIT, Rohini, New Delhi.
2. Wheeler D., and R. Needham. TEA, a Tiny Encryption Algorithm, Proceedings of the 2nd International Workshop on Fast Software Encryption, Springer-Verlag,1995,pp.97-110.
3. Anjula Gupta,Navpreet Kaur Walia, " Cryptography Algorithms: A Review", Volume 2, Issue 2, International Journal of Engineering Development and Research, © 2014 IJEDR.
4. P. Israsena, "Design and Implementation of Low Power Hardware Encryption for Low Cost Secure RFID Using TEA", 2005 [Fifth International Conference in Information, Communications and Signal Processing, Bangkok, Thailand].
5. Massey, J.L, "An Introduction to Contemporary Cryptology", Proceedings of the IEEE, Special Section on Cryptography, 533-549, May 1988.
6. John Kelsey, Bruce Schneier, and David Wagner, "Related-key cryptanalysis of 3-Way, Biham-DES, CAST, DES-X NewDES, RC2, and TEA," LNCS, Vol. 1334, pp. 233-246, Springer-Verlag 1997.
7. The Sampling of Noise for Random Number Generation: Craig S. Petrie and J. Alvin Connelly, "The Sampling of Noise for Random Number Generation", IEEE Proceedings of ISCAS, 6:26-29, 1999.
8. PIC 24FJ128GA204 Datasheet available at www.microchip.com/PIC24FJ128GA204/
9. Senthil Kumar.S, Manjupriya.M, "Microcontroller based Cryptosystem with Key Generation unit," International Conference on Computer, Communication and Electrical Technology, pp. 1-7, March, 2011
10. Bruce Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, 2002