Journal of Engineering Science and Technology
Vol. 8, No. 4 (2013) 406 - 415

A LOW COST IMPLEMENTATION OF MODIFIED ADVANCED ENCRYPTION STANDARD ALGORITHM USING 8085A MICROPROCESSOR

SALIM M. WADI[1,2,*], NASHARUDDIN ZAINAL[1]

[1]Department of Electrical, Electronic & Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
[2]Communications Technical Engineering Dept., Najaf Technical College, Foundation of Technical Education, Bagdad, Iraq
*Corresponding Author: salimmw@eng.ukm.my

Abstract

The high security communication systems became an urgent need in recent years for both governments and peoples desiring protection from signal interception. Advanced Encryption Standard (AES) is a famous block encryption algorithm which has several advantages in data encryption. However, AES suffer from some drawbacks such as high computations, pattern appearance if apply for image encryption, and more hardware requirements. These problems are more complicated when the AES algorithm is used for multimedia encryption. New modification to AES-128 algorithm which reduce the computations and hardware requirements are proposed by enforcement Mixcolumn transformation in five rounds instead of nine rounds as in original AES-128. Second proposed is suggest new simple S-box used for encryption and decryption. The implementation of advanced encryption standard algorithm is important requirement where many researches proposed different items to this purpose. A simply item proposed in this paper to speedy, low cost implementation of Modified Advanced Encryption Standard (MAES) cryptographic algorithm is 8085A microprocessor. The results prove that the modifications of AES make implementation it by 8085A microprocessor more effective.

Keywords: AES, Encryption time, Reduced round attacks, 8085 processor.

1. Introduction

The rapid evolution in communication systems such as satellite, mobile network, internet, and earth communications produced important need to protect and

preserve sensitive and critical public, private, and national infrastructures and their respected data against attacker and illegal copying and distribution [1]. Cryptography provides a method for securing and authenticating the transmission of information over insecure channels [2]. One of the cryptography algorithms used widely in most applications such as smart card, cell phone, automated teller machines, and www servers is the Advanced Encryption Standard (AES) [3]. AES is very strong against resist attacks because it has long key. AES algorithm has several advantages such as reliable, flexible, and compatible with hardware implementation. However, it suffers from some drawbacks; for example, encryption time is long, especially with multimedia encryption [4].

Many modifications are proposed on AES algorithm by change the method of S-box construction [5, 6], modify the Mixcolomun transformation [7], or by used Chaotic Henon map and Arnold's cat map to generate the AES key and shuffled the image pixels [8, 9]. All these modifications to AES didn't reduce the time of encryption and still AES has high computations. In this paper, new modification of AES is proposed by reducing the number of Mixcolumn transformation executions from nine to five to reduce the overall execution time of AES. Also, new simple S-box is proposed with property that using same S-box for encryption and decryption.

The implementations of AES are carryout into two ways, by FPGA or microcontroller. Orlando, J. et al. [10] and Gielata et al. [11] used FPGA to implement AES algorithm in different ways. Hyubgun Lee et al. [12] proposed the sensor network with high security to analyse the communication efficiency through performance evaluation of AES ciphering system depending on data length, and cost of operation per hop according to the network scale. The authors conclude that if the scale of the sensor network increased, this leads to double the delay as well as increasing in the energy disbursed. Kai Schramm et al. [13] proposed lab to worthily carryout the Advanced Encryption Standard (AES) on a smart card using Atmel ATMega163 Reduced Instruction Set Computer (RISC) microcontroller in assembly.

An implementation of MAES algorithm based on simple item with low cost is proceeding of in this paper. 8085A microprocessor as processing tool with very effective manner and reasonable speed is used in this implementation.

The rest of paper is organized as, in Section 2 initial AES algorithm is presented, and an elucidation of modification proposed of initial AES presented in Section 3. Section 4 explains the proposed implementation. Results and conclusion are shown in Sections 5 and 6 respectively.

2. AES Algorithm

After breaking Data Encryption Standard (DES) by attackers, the National Institute of Standards and Technology (NIST) held a series of conferences to choose new encryption algorithm with implementation property in software, firmware, hardware, or any combination thereof. The advanced encryption standard (AES) which is developed by two cryptographers, Joan Daemen and Vincent Rijmen have been selected by NIST as standard ciphering algorithm in 2002 [14].

There are three versions from AES algorithm depending on length of the key (AES128, AES192, and AES256) bit. These different length keys are arranged in matrices with sizes of 4x4, 4x6, and 4x8 respectively and 128 bit block data which constructed in 4x4 matrix called state [15]. AES algorithm is divided into four sequential operations where these operations are made on a state with (10, 12, 14) rounds based on key length as shown in Fig. 1. Details of AES algorithm operation are shown in sub section below:

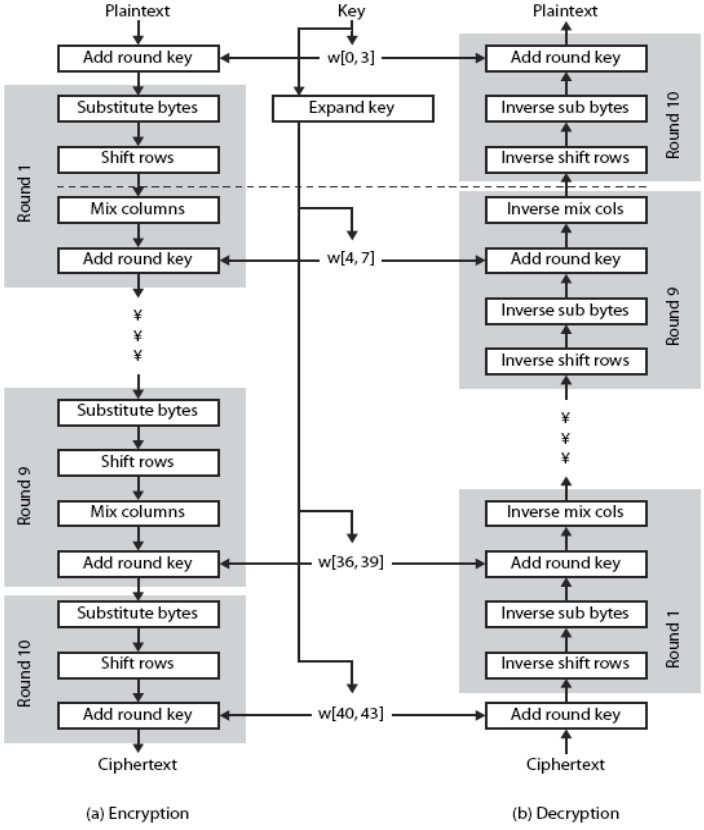

Fig. 1. AES Structure: (a) Encryption Operation, (b) Decryption Operation.

2.1. AES steps round

2.1.1. SubByte transformation

SubByte operation is a nonlinear byte substitution that substitutes the state bytes independently using substitution table (S-box). S-box is constructed by taking the multiplicative inverse in the finite field GF(28) as in Eq. (1), then apply the affine transformation (over GF(2)) [14]:

$$b_i' = b_i \oplus b_{(i+4)mod8} \oplus b_{(i+5)mod8} \oplus b_{(i+6)mod8} \oplus b_{(i+7)mod8} \oplus c_i \quad (1)$$

where $0 \leq i < 8$, b_i is ith bit , c_i is the ith bit of a byte c with value {01100011}.

Equation (2) below represent S-box affine transformation element in matrix form [14]:

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\;\;0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{2}$$

2.1.2. ShiftRow transformation

Shifting operation applies on state rows in this step, where 1st row no shifted, 2nd row shifted to right one time, 3rd row shifted to right two times, and 4th row shifted to right three times. Figure 2 illustrates the ShiftRow transformation [14].

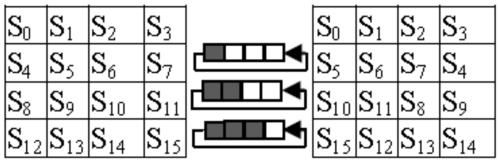

Fig. 2. ShiftRow Operation.

2.1.3. MixColumn transformation

MixColumn transformation carries out on state column by column. Each byte is replaced by a value dependent on all 4 bytes in same column through multiplication state matrix in GF(28) as in Eq. (4) using prime poly as shown in Eq. (3) [15]:

$$m(x) = x^8 + x^4 + x^3 + x + 1 \tag{3}$$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} \tag{4}$$

2.1.4. AddRoundKey transformations

Final operation in AES round is the AddRoundKey (ARK) transformation. ARK transformation is a simple bitwise XOR between state matrix and sub key as shown in Fig. 3.

Fig. 3. AddRoundKey Transformation.

2.2. AES key expansion

The AES key expansion operation takes as input a 4-word (16-byte) initial key and produces a linear array of words, providing a 4-word round key for the initial AddRoundKey stage and each of the 10/12/14 rounds of the cipher. It involves copying initial key into the first group of 4 words, and then constructing subsequent groups of 4 words for each group depend on the values of the previous group. The first word in each sub key gets special dealing with rotate + S_box + Rcon on the previous word before XOR'ing with first word from next sub key, for more details see [14]. Figure 4 shows an example of key expansion operation for first sub key.

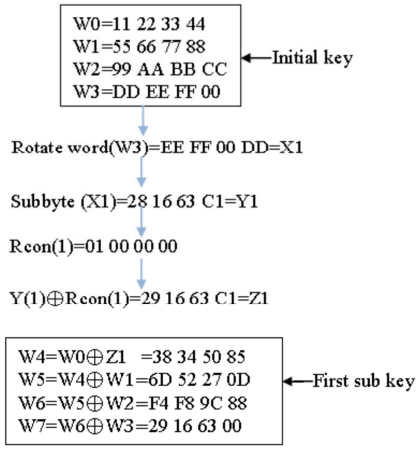

Fig. 4. Key Schedule Example.

3. Modified Version of AES Algorithm

AES algorithm is a block cipher algorithm that means execute number of transformations more than one times (in iterations). The bigger challenge facing the use of any ciphering algorithm is the encryption time especially because with multimedia application. The attack is another problem to all ciphering systems including AES. Therefore, we should do any modification to ciphering system carefully such that it does not give any lacuna to adversary. One of important types of attacks named as reduced round attacks. Theoretically, the adversary can broke AES algorithm for 7 round with 2^{120} encryption operations [16]. Therefore,

we cannot reduce the rounds of AES algorithm in order to decrease encryption time because this makes AES weak against this types of attacks. AddRoundKey is one transformation of AES round where X-OR operation between new sub key derived from previous sub key and state will be carrying out. This is yet another reason to prevent us from reducing the number of AES rounds.

3.1. Mixcolumn reduction

For previous two reasons we think by diminishing one of AES round transformations. MixColumn transformation has large amount of computations compared with other transformation of AES (SubByte, ShiftRow, and AddRoundKey). Therefore, we suggest reduce the number of times of MixColumn execution to 5 times rather than 10. This will decrease the computation amount in factor RF calculated in Eq. (5)

$$RF = 4 * MCC * (DB/16) \tag{5}$$

where *RF* is reduction factor, *MCC* is number of arithmetic operations in Mixcolumn transformation, and *DB* is total data bytes.

3.2. New S-box

New and simple S-box was proposed using Eq. (6)

$$x(i,j) = i\,F - j \tag{6}$$

where *x(i,j)* is element value in proposed S-box with location determined by *(i,j); i, j,* and *F* are Hexadecimal numbers.

The proposed S-box matrix is constructed simply. In addition, one S-box used to encryption and decryption (in SubByte and Inverse SubByte operations) instead of using two matrices, one for SubByte and the other for Inverse SubByte as in initial AES where this will reduce the hardware requirements.

4. MAES Implementation

The best encryption algorithm that is easy to implement in software and in hardware [10]. Inventors of AES algorithm designed it with an idea in mind of ability for its efficient execution using different platform such as CPU, ASICs, and FPGA. Microprocessor is a cheap, readily programmed, and has high efficiency. For these reasons, we choose the microprocessor to implement the MAES algorithm.

Same items used in previous our work produced in [17] will be used in this paper to implement the MAES. Figure 5 shows a block diagram to proposed system used to implementation:

The system consists from the following parts:

- 8085A microprocessor: this processor is from Intel family, 8 bit processor, operating frequency is 5 MHz.
- EEPROM: about 0.978 Kbyte used to save the operating programs and S Box and inverse S Box.

- RAM: about 0.76 Kbyte
- Keyboard and 7-sigment display to input the plaintext and key block as inputs and display the cipher text as outputs.
- I/O ports.

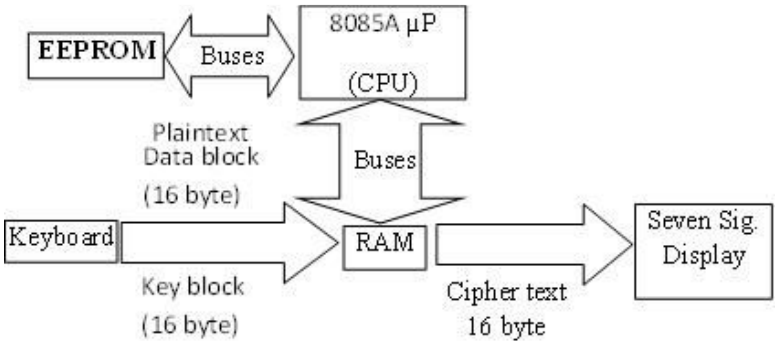

Fig. 5. System Block Diagram.

5. Results and Discussion

Performance of implementation system to MAES algorithm is evaluated through:

- Simplicity:
  The proposed implementation is very simple as hardware and the programming requirement is very easy where assembly language used to write the essential programs to employment the microprocessor.
- Cost:
  As known to all, the price of 8085 microprocessor IC is very low about 1-1.5 $ which is the main part of suggested circuit, therefore we think that the cost of the complete circuit is very low compared with other implementation using FPGA or microcontrollers.
- Speedy:
  Two factors contribute to getting the speedy implementation to encryption algorithm with high security. These factors are modifications proposed to AES algorithm and system suggested to implements the MAES algorithm.

The specifications of suggested circuit and time required to encryption one block data using MAES are shown in Table 1.

Table 2 shows the implementation results to the circuit. It can be seen from this table that the speed of complete implementation is reasonable and compatible most applications as example to researches purpose.

To demonstrate the effectiveness of proposed implementation, we compare it with other three implementations, which used the microcontroller and microprocessor. The evaluation comparison is shown in Table 2, where the

proposed implementation is compared with those proposed in previous work [12, 13, 17].

Table 2 shows that the consumption time in proposed method little than time required in all three other works. Therefore, the proposed method is effective in terms of cost, time consumption, simplicity, and required peripherals such as memory.

Table 1. The Circuit Specifications and Time Execution for One Block.

Type of Processor	8085A µP
Frequency	5 MHz
Time of 1Block Data Encryption	8.56 ms
No. of Exec. Cycles (1 data block 1 round)	4280
EEPROM Needed	0.722 Kbyte (contain the program memory)
Volatile Memory	0.76 Kbyte
Internal Registers	7 (8-bit), 1 (16-bit)

Table 2. Comparison of Suggested Implementation Results with Proposed Implementations in [12, 13, and 17].

Implementation of AES	Current	Proposed in [17]	Proposed in [12]	Proposed in [13]
Type of CPU	8085A µP	8085A µP	ATmega644p µC	ATMega163 µC
Frequency	5 MHz	5 MHz	12 MHz	3.57 MHz
Time of encryption	8.56ms	17.66ms	449 ms	22.50 ms
EEPROM	0.722 Kbyte	0.978 Kbyte	8 Kbyte	32 byte
RAM	0.76Kbyte	0.76Kbyte	Not found	2.234 Kbyte
Internal registers	7(8-bit), 1(16-bit)	7(8-bit), 1(16-bit)	Not found	32 (8-bit)

6. Conclusions

One of the popular and effective encryption algorithms is the Advanced Encryption Standard (AES). However, AES suffers from some disadvantages, for example, need for more computations and then long encryption time. Some work proposed to modify AES algorithm to increase the security level or to disserting some problems which appear when AES is applied to image encryption. Papers which address the encryption time of AES almost non-existent.

In this paper some modifications are proposed on AES to reduce the encryption time with conservation the security level. Mixcolumn steps need large amount of computations. First modification is decrease the number of times of execution the Mixcolumn transformation from 9 to 5 where this doesn't affect much on security level. Another modification is replacement the initial S-box with new one. The advantages of new S-box are easy and use same to encryption and decryption.

A number of systems are suggested to implement AES algorithm using FPGA and microcontroller. FPGA is high speed items but it is expensive and more complicated. The implementation systems that depend on microcontroller are slow and also expensive compared with microprocessor. A new low cost implementation system based on 8085A microprocessor was proposed in this paper. Results showed that the proposed method to implement modified AES is very effectives in terms of cost and speed with high security in compared with other three implementations. As a result the time and cost of the proposed implementation is good in compare with other implementation.

References

1. Lian, S.; Sun, J.; and Wang, Z. (2005). Security analysis of a chaos- based image encryption algorithm. *Physica A: Statistical Mechanics and its Applications*, 351(2-4), 645-661.
2. Daemen, J.; and Rijmen, V. (2002). *The design of AES- The advance encryption standard*. Springer-Verlag.
3. Borujeni, S.E.; and Eshghi, M. (2011). Chaotic image encryption system using phase magnitude transformation and pixel substitution. *Telecommunication Systems*, 52(2), 525-537.
4. Huang, C.-W.; Yen, C.-L.; Chiang, C.-H.; Chang, K.-H.; and Chang, C.-J. (2010). The five modes AES applications in sounds and images. *Sixth International Conference on Information Assurance and Security (IAS)*, Atlanta, GA, 28-31.
5. Telagarapu, P.; Biswal, B.; and Guntuku, V.S. (2011). Security of image in multimedia applications. *International Conference on Energy, Automation, and Signal*, 1-5.
6. Jing, M.-H.; Chen, J.-H.; and Chen, Z.-H. (2008). Diversified Mixcolumn transformation of AES. *International Conference on Information, Communication & Signal Processing*, 1-3.
7. Chen, Y.-C.; Zou, X.-C.; Liu, Z.-L.; Chen, X.-F.;and Han, Y. (2008). Dynamic inhomogeneous S-Boxes design for efficient AES masking mechanisms. *The Journal of China Universities of Posts and Telecommunications*, 15(2), 72-76.
8. Bin Muhaya, F.T. (2013). Chaotic and AES cryptosystem for satellite imagery. *Telecommunication Systems*, 52(2), 573-581.
9. Tran, M.T.; Bui, D.K.; and Duong, A.D. (2008). Gray S-box for advanced encryption standard. *International Conference on Computational Intelligence and Security*, 253-258.
10. Hernandez, O.J.; Sodon, T.; Adel, M.; and Kupp, N. (2008). A low cost advanced encryption standard (AES) Co-processor implementation. *Journal of Computer Science and Technology*, 8(1), 8-14.
11. Gielata, A.; Russek, P.; and Wiatr, K. (2008). AES hardware implementation in FPGA for algorithm acceleration purpose. *Proceedings of International Conference on Signals and Electronic Systems*, 137-140.

12. Lee, H.; Lee, K.; and Shin, Y. (2009). AES implementation and performance evaluation on 8-bit microcontrollers. *International Journal of Computer Science and Information Security*, 6(1), 70-74.
13. Schramm, K.; and Paar,C. (2004). IT security project: Implementation of the advanced encryption standard (AES) on a smart card. *Proceedings of International Conference on Information Technology: Coding and Computing*, 1, 176-180.
14. Federal Information Processing Standards Publication 197 (2001). Announcing the Advanced Encryption Standard (AES).
15. Rais, M.H.; and Qasim, S.M. (2009). A novel FPGA implementation of AES-128 using reduced residue of prime numbers based S-Box. *International Journal of Computer Science and Network Security* (*IJCSNS*), 9(9), 305-309.
16. Bouillaguet, C.; Derbez, P.; Dunkelman, O.; Fouque, P.; Keller, N.; and Rijmen, V. (2012). Low-data complexity attacks on AES. *IEEE Transactions on Information Theory*, 58(11), 7002-7017.
17. Wadi, S.M., and Zianal, N. (2012). A low cost implementation of advanced encryption standard algorithm using 8085A microprocessor. *Proceedings of 3rd International Technical Conference* 2012, 157-163, KL, Malaysia.