An Improvement of Advanced Encryption Standard

Md. Shafiul Alam Forhad[1], Siam Riaz[1], Md. Sabir Hossain[1], Mrinmoy Das[2]
[1]Department of Computer Science & Engineering, Chittagong University of Engineering & Technology, Bangladesh
[2]Department of Computer Science and Engineering, Bangladesh University of Business and Technology , Bangladesh

Summary

A modified version of the Advanced Encryption Standard algorithm is developed here for the aim of security and easier implementation. As the cloud users and communications over the network are increasing tremendously and therefore the users also would like faster storing and access to data on the cloud, it's become the key issue to provide security. Advanced Encryption Standard (AES) is a vastly used technique for providing security. This algorithm is modified to make it efficient but some security needs to be compromised. Here a different key generation process is used. Transpose is introduced and MixColumn process is eliminated for removing complex calculation. Thus, it provides a less complex technique as compared to the main algorithm. It will be easily implemented within the cloud as well as for hiding info in the personal computer.

Key words:
Advanced Encryption Standard, Transposition, Key expansion

1. Introduction

Cryptography is concerned with message confidentiality, conversion of plaintexts into an inconceivable one and making it unintelligible for the eavesdroppers without the knowledge of the secret key which is used to ensure security. Security is very important in many areas like communication or storing data on clouds. Cryptographic methods provide a way to secure data against unauthorized access. Nowadays many cryptographic algorithms have been developed for better security. Here we have used modified AES (Advanced Encryption Standard). AES has rapidly used encryption in recent times. AES is constituted of some bit substitutions, permutations, round key functions and transformations [6]. AES is a faster algorithm and it is not breakable to all practical attack exists right now. Thus, AES is still one of the most reliable encryption standards for advanced security of information. For making the calculation easy some modifications are made to increase the efficiency of the algorithm.

When the evaluation of the improved advanced encryption standard is discussed, especially from the methodological viewpoint, the essential points are given below:

(i) For encryption of plaintext, a secret key is used by symmetric key algorithms and in case of decryption none other than the same key is needed for obtaining the plaintext.

(ii) Here symmetric key algorithm is selected because of comparatively high speed and computational efficiency to control a huge amount of data. In symmetric cryptosystems, the strength of algorithm rests in the key length. If the length of the key is large enough, then encryption will be stronger.

(iii) Generating keys is very important in encryption methods. The keys are generated as the way thus the eavesdroppers can’t predict anything about the original key.

(iv) For AES-128 there are total 10 rounds. Thus, the total of 10 round keys need to be generated from the cipher key.

(v) The MixColumn is replaced with transposition. Thus, it reduces the computational complexity of the algorithm and easier to implement. It also provides faster encryption or decryption as compared to the original algorithm. These modifications are made for the better performance of the algorithm.

2. Background

Stream cipher and block cipher are two types of symmetric key ciphers. Advanced Encryption supports variety of key lengths like 128 bits, 192 bits and 256 bits. Strength of AES is very much dependent on key size [1], [6]. AES-128 needs total ten rounds; AES-192 needs total twelve rounds and AES-256 needs fourteen rounds. There are total 10 rounds for AES-128. From the initial key, 10 more round keys are generated which is applied to each round. Each key is divided into four words. So, total of 44 words are needed. 4 words or total 16 bytes are arranged in a matrix order of 4X4 [6]. First of all, logical X-OR operation is performed between cipher key and plaintext. Then, in each round there are several types of operations needs to be performed which are described in the following-

(i) SubBytes: The values of each column and row elements are substituted form a fixed table known as S-Box. After substitution, it constructs a 4X4 matrix.

(ii) ShiftRows: Cyclically byte shifting operations are held here. First row of the matrix is unchanged. 1st, 2nd and 3rd positions shifted to the left for 2nd, 3rd , and 4th row respectively.

(iii) MixColumns: Each column of the matrix after shift operations are multiplied by a predefined matrix. Thus the way each column is transformed and a new 4X4 matrix is formed.

Manuscript received November 5, 2018
Manuscript revised November 20, 2018

(iv) AddRoundKey: The matrix constructed after MixColumns operation which represents 16 bytes are treated as 128 bits. At each round, a round key is added after the MixColumns operation.

In each round these four operations described above are performed. But in the final round, MixColumns is not needed. To obtain the ciphertext, after shift operation it is added to the last round key.

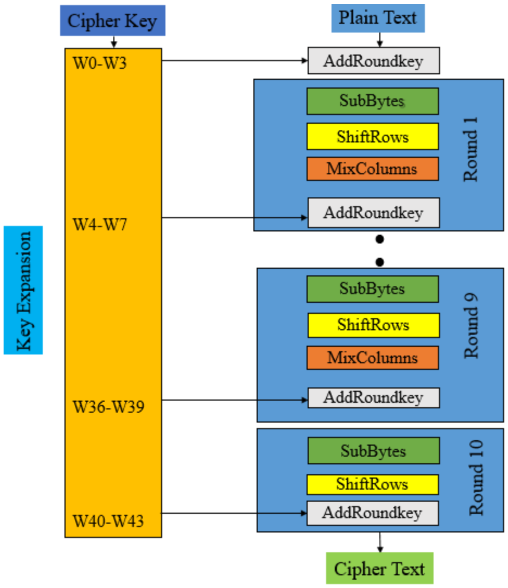


Fig. 1 Encryption Process of Advanced Encryption Standard [1]

3. Related works

In [1], a simple method to secure data is proposed where encryption of data is performed using Advanced Encryption Standard (AES). This encryption approach to protect data is explained over making it unpredictable for attackers. In case of security of data, implementation of AES provides the advantage of minimal space requirement and less calculation time than other algorithms.

In [2], a system is proposed treating security as a major concern which focuses on providing protection of data while transferring using the encryption method. The security of data and access of data by the third-party servers is focused by this method.

In [3], here a modified version of AES is proposed. These modifications are applied to the round operations of the algorithm. The complexity of the encryption method increased by these types of changes. Thus, it becomes difficult for the adversary to predict a pattern of the key. This proposed method can be applied without enhancing the block size of the key. The modifications proposed in the algorithm will help to encrypt the data by making stronger diffusion and confusion. It makes the algorithm more complex compared to the older one.

In [4], a method for security of data storage in cloud computing is presented which gives more sustainable privacy. Though the execution time is extended it preserves the privacy where the symmetric encryption algorithm AES is used to reserve data. General analysis of the experiment shows that the method provides strong security and efficiency is high. The system can provide high security for a commercial public cloud.

In [5], encryption model known as AES algorithm is the more prominent cryptographic algorithm. For breaking an encryption algorithm, huge time is required and it is dependent on the key size. AES allows a variety of versions depending on key sizes. Here RSA and AES algorithm are used to make the verification process more effective and flexible.

In [6], here AES is used to deliver a larger dimension of privacy and verification for making the database more invulnerable. AES is more resistant against different types of attacks like key attack, square attack, differential attack, and other practical attacks. Thus, the AES algorithm remains one of the most used encryption techniques. Application and data will be protected furthermore against future attacks like smash attacks. Memory consumption of AES algorithm is low but provides faster performance without any weaknesses and limitations. But in case of other symmetric algorithms, they have some weaknesses and not provide as high performance as AES and storage space is needed more as compared to AES.

In [7], here security issues related to cloud storage and the challenges met by the third-party service provider are discussed and the study of different types of security algorithms is presented. The experimental result shows that the AES algorithm takes minimum time to encrypt and RSA needs longer encryption time. The final simulation outcome shows that the AES algorithm provides overall better performance.

In [8], a method is applied to overcome the computational overhead, the Advanced Encryption Standard (AES) is modified after analyzing it. It is modified in such a way that the calculation of the algorithm is minimized and the performance of the algorithm is improved. The Mixcolumn step is eliminated and the permutation step is introduced. The modified version of this algorithm provides faster encryption technique for protection of multimedia data. This lightweight encryption algorithm is compatible with large plaintext and transmits of an image.

By implementing the approach in [9], the complexity of the architecture can be reduced. This method can provide a large scale of the data rate for both the process of encryption or decryption. This method is suitable for a smart card, smartphones or implementing other hardware related difficult applications.

In [10], a performance comparison has been conducted among the six general cryptographic algorithms known as DES, AES, 3DES, Blowfish, RC2 and RC6 at various kinds of settings for each type of algorithm like different block size, various data types, length of key, time needed to encryption or decryption process. The outcomes from the experiments are provided to evaluate how effective these algorithms are in different settings.

In [11], 512-bits key size version of AES and suitable necessary hardware to implement the algorithm are proposed. The key size of 512-bits used by this new approach enables it to make it more prohibitive to attackers when they try to break the system. AES-512 can be implemented for applications where a huge amount of security is required. AES-512 provides a huge increment in throughput of 230% compared to AES-128. The huge length of the key of this method provides more security, and the throughput is enhanced due to its larger key size.

In [12], errors on the implementation of hardware using AES and the effects by examining the extension of these errors to the outputs is explained. Redundancy-based scheme and a scheme which uses an error checking code are two fault detection techniques which are proposed here. Cost of hardware and detection faults of these two schemes are determined.

In [13], a technique for implementing logical functions of S-Box is proposed. Only one-quarter of the gates is required as compared to other fast techniques. Thus, it is quite beneficial. A differential output can be generated that enables to make faster implementation of exclusive-OR logic. The experiment result shows that this proposed method is both fast and area efficient.

In [14], here some modern encryption algorithms are discussed which are now used in many systems. Basically, these algorithms concentrate on different types of existing encryption methods and differences and similarities among them. Image and information encryption are also explained. Finally, the result of the survey shows that all approaches are suitable for real-time encryption but each method has different characteristics. From this analysis, it can be determined which method is useful in different applications.

In [15], security of cloud data storage is focused on to ensure quality service to the user. An effective method having two consequential features is proposed to make sure of user data on cloud storage to be correct. By using the same token for authentication of rejected data, cloud storage faultlessness or localization of data error are executed by the system to find the identification of misbehaving server. The method ensures user data correction in cloud data storage. The system also supports dynamic data where operations like delete, update etc. can be performed.

In [16], a review on various encryption algorithms is presented to increase the data security to avoid the security issue. To enhance data security, different types of cryptographic algorithms are available. Security of the cloud depends on trusted computing and cryptography. A review on RSA, AES encryption algorithms and a performance comparison between them are presented here by using the cryptographic methods for increasing security of data in cloud data storage.

In [17], Advanced Encryption Standard (AES) is now largely used in wireless communications due to lower memory consumption and fast speed of data encryption. An improved AES algorithm is proposed here to overcome the deficiency of the key expansion method which key is usually attacked by square. The double S-box model with non-linear structure is employed in this algorithm. Experimental results indicate that this AES algorithm can improve the security of the key in the same condition of algorithm efficiency.

4. The Proposed Method

In this section, the encryption and decryption processes using our proposed AES algorithm are described in details.

4.1 Encryption Process

Steps of the proposed encryption algorithm are described in the following-

1) First of all, the key is divided into 4 words. Then it is expanded into 44 words to generate the round keys. Total 11 keys including the initial key are generated.
2) The plaintext is X-ORed with the initial round key.
3) After X-OR operation, it is arranged into a 4X4 matrix. Total 16 bytes are represented by the matrix. Each column of the matrix represents a word. Then each element of the matrix which represents a byte is substituted according to the value of lookup table (S-Box). This is SubByte operation.
4) Then ShiftRow operation is implemented on the resultant matrix. The bytes are cyclically shifted. The first row of the matrix is unchanged. 1, 2 and 3 positions are shifted to the left respectively in case of second, third and fourth row. The fall up byte is stored into the right in order.
5) The resultant matrix from the ShiftRow operation is transposed.
6) After transpose operation, the matrix is arranged into 16 bytes or 128 bits and it is X-ORed with the next round key.
7) Then again, these operations are implemented described in 2,3,4,5 and 6 step till the round Nr-1.

Here, Nr denotes the no. of rounds. In case of AES-128, Nr=10.

8) In the final round (Nr), SubBytes and ShiftRow operations are implemented but transposition operation needs not to be applied. After ShiftRow operation the resultant matrix is arranged into 128 bits and X-ORed with the final round key. This resultant 128 bit is the ciphertext of the plaintext.

In fig.2, the encryption process is shown step by step with round operation.

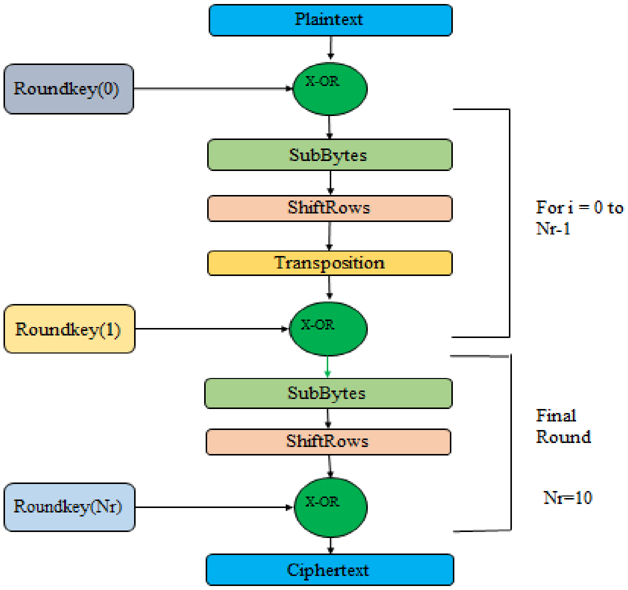


Fig. 2 Flowchart of Proposed Algorithm

4.1.1 Key Expansion

The 128-bit key is expanded into subkeys. Total 11 subkeys are needed including the initial key. Each key is divided into 4 words. It is expanded into 44 words according to the fig.3. 4 words or 16 bytes are arranged into a 4X4 matrix. Each column represents a word and four words represent a key. For the first key, it is arranged as follows-

Key0=W0 W1 W2 W3

Now it is expanded to a total 44 words [17].

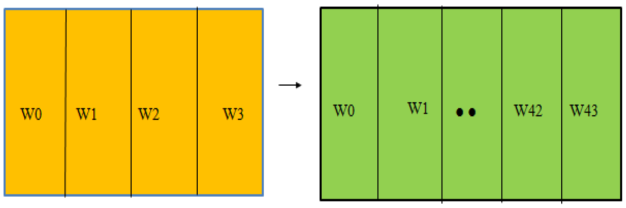


Fig. 3 Key Expansion

Rcon (1) = W0 ⊕ W3

G1 = Rcon (1) ⊕ SubNib (RotNibW3)

- Rcon refers to round constant which is generated by XORing the first and last word of each round key.
- The 4 bytes are cyclically shifted one position to the left. This is rotation of nibble bit.
- Then for each byte, the values are substituted from S-Box.

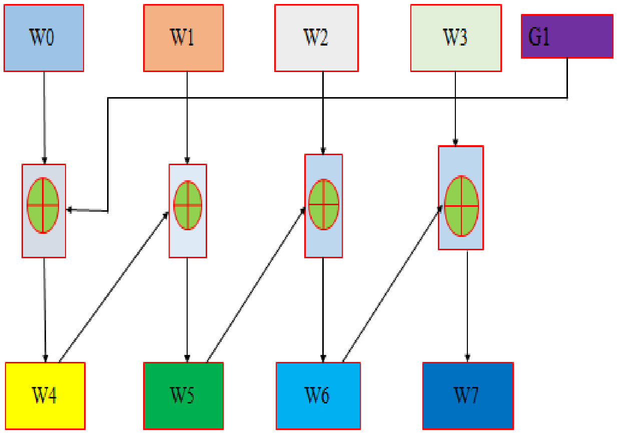


Fig. 4 Key Generation Process [17]

From the fig.4 we can find the following:

W4= The XOR between W0 and G1

W5= The XOR between W1 and W4

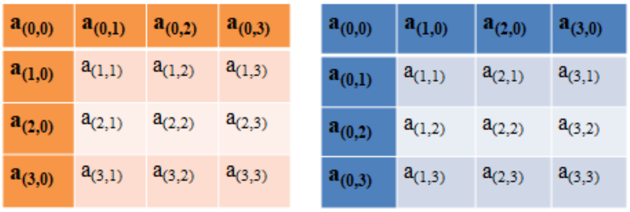
W6= The XOR between W2 and W5

W7= The XOR between W3 and W6

Key1=W4 W5 W6 W7

Thus the way all words are generated. The next 4 words represent Key2 and finally, we get all the round keys.

4.1.2 Transposition



$a_{(0,0)}$	$a_{(0,1)}$	$a_{(0,2)}$	$a_{(0,3)}$
$a_{(1,0)}$	$a_{(1,1)}$	$a_{(1,2)}$	$a_{(1,3)}$
$a_{(2,0)}$	$a_{(2,1)}$	$a_{(2,2)}$	$a_{(2,3)}$
$a_{(3,0)}$	$a_{(3,1)}$	$a_{(3,2)}$	$a_{(3,3)}$

$a_{(0,0)}$	$a_{(1,0)}$	$a_{(2,0)}$	$a_{(3,0)}$
$a_{(0,1)}$	$a_{(1,1)}$	$a_{(2,1)}$	$a_{(3,1)}$
$a_{(0,2)}$	$a_{(1,2)}$	$a_{(2,2)}$	$a_{(3,2)}$
$a_{(0,3)}$	$a_{(1,3)}$	$a_{(2,3)}$	$a_{(3,3)}$

Fig. 5 Transpose Operation

Transpose of the matrix is shown in the fig.5. The operation is performed by switching row and column indices of the matrix. MixColumn operation is replaced by this operation. It results in less time requirement and simplicity of calculation. After ShiftRow operation, the resultant matrix is transposed. After transposing a matrix, if it is again transposed, we will get the initial matrix. In this case, column bytes become row bytes and vice-versa.

4.2 Decryption Process

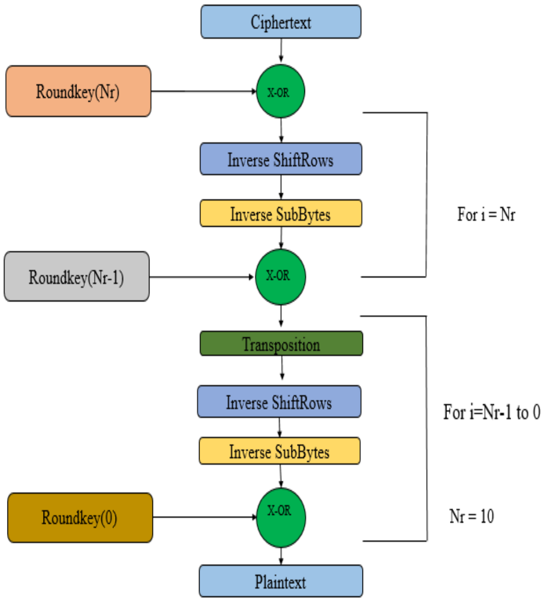


Fig. 6 Decryption Process

The decryption process of our proposed algorithm is shown in fig. 6. The steps required to decrypt the cipher text to get the plain text are provided below:

1) The ciphertext is XORed with the round Key10 and arranged in a 4X4 matrix.
2) Then inverse ShiftRow operation occurs. First row remains the same. In case of the 2nd, 3rd and 4th row, 1, 2 and 3 positions shifted to the right and the fall up elements are stored in the left.
3) After inverse ShiftRow, the values of each byte are substituted according to the inverse SubBytes from the S-Box inversely.
4) Then the resultant matrix is transposed.
5) The resultant matrix after inverse SubBytes are now arranged in 128 bits and XORed with Key9. Then 2, 3 and 4 occurs. In each round after performing these operations, the keys are XORed in descending order like Key9, Key8, and Key7 etc.
6) Finally, when key0 is applied for XOR operation with the resultant values from the inverse subBytes we will get the desired plain text.

5. Experimental Findings

5.1 Result by Simulation

Te experimental outcome shows that modified AES is quite faster than the original AES. Because the most complex part is eliminated or easier implementation and faster encryption or decryption process. It reduces the complex calculation and makes the implementation simple. The result of the average run time of two programs is shown in Figure 7.

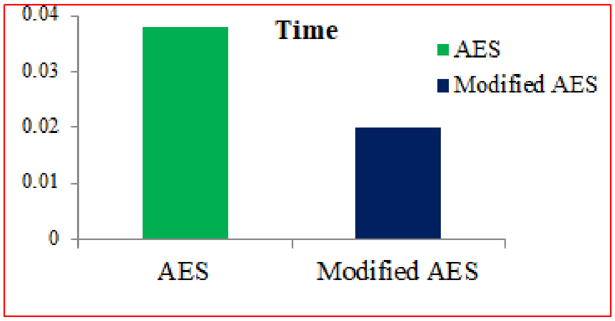


Fig. 7 Run Time comparison of AES and Modified AES

5.2 Comparative Analysis

A modified version of AES was previously proposed where the MixColumn operation is eliminated and permutation is introduced which is 26.75 % more efficient than the original one [8]. Our proposed method is 47.37% more efficient than the original AES.

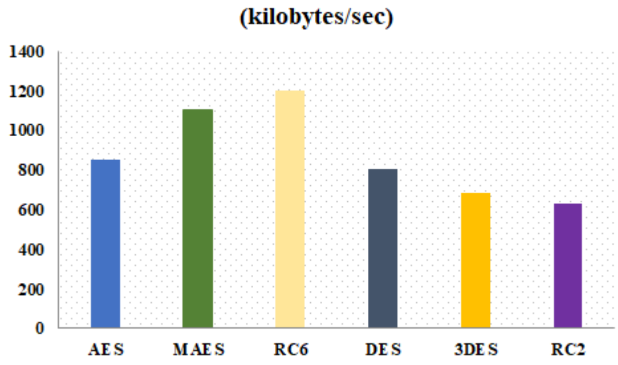
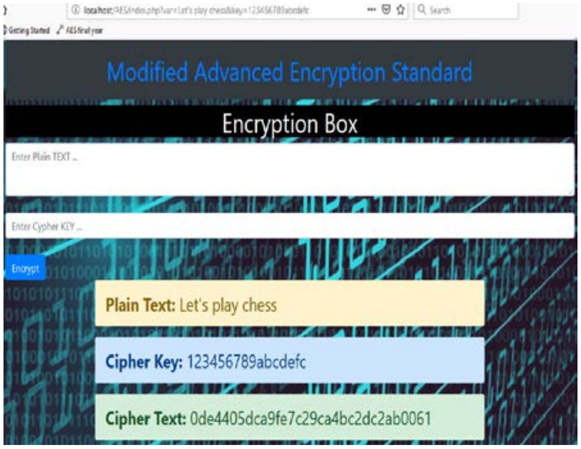


Fig. 8 Throughput of each encryption algorithm

From Fig.8, we can see that modified AES is much faster in case of encryption only RC6 provides better encryption time but as it is a stream cipher it can't take a block of plaintext. But as compared to other algorithms like DES, 3DES, RC2 AES provides better encryption of data. Modified AES provides quite faster encryption as compared to the original AES and other algorithms. 3DES provides a good amount of security but the encryption is quite slow. The only RC6 is faster from the modified algorithm.

6. Results and Discussion

Modified AES is based on AES-128. The cipher of N bits key size has a key space of entire 2^N bytes. Thus there are total 2^128 = 3.4 * 10 ^ 38 alternative keys. Minimum 5.4*10^18 years are needed to get the exact key [18]. Because of the large key space and highly complex calculation, the security of AES is not threatened by the brute force attacks. The most significant cryptanalysis of AES is biclique attack but it is not a great improvement over brute force attack [18].
The complexity of computation is 2^126. 18 as compared to 2^ 128 of brute-force. 1.5* 10^18 years are required to obtain which is not feasible to perform [18]. Here in this proposed method, the MixColumn has been replaced by transposition. Though the MixColumn provides more security it makes the encryption slow [8]. Thus, to reduce the complexity of MixColumn, here Transposition is used. It makes a simple implementation of the algorithm. Thus, it provides a much better encryption time.
The interface for encryption is shown in the fig.9. Here plaintext and cipher key have to be taken as input and output will be the ciphertext. In fig.10, the decryption interface is shown. Here the ciphertext and same cipher key need to be taken as input and it will give the decrypted text that is the plaintext.



Modified Advanced Encryption Standard

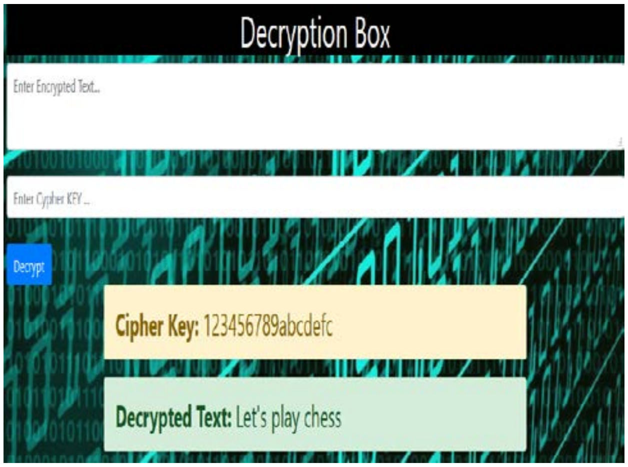
Encryption Box

Plain Text: Let's play chess

Cipher Key: 123456789abcdefc

Cipher Text: 0de4405dca9fe7c29ca4bc2dc2ab0061

Fig. 9 Output of the Experiment (Encryption)



Decryption Box

Cipher Key: 123456789abcdefc

Decrypted Text: Let's play chess

Fig. 10 Output of the Experiment (Decryption)

7. Conclusion

Few theoretical attacks can be possible against AES, however, all of them are inefficient to execute. Though the computers are becoming double powerful every 1.5 years it would still need a long time before AES becomes breakable. The most meaningful security issue is the side-channel attacks [18]. While it's not possible to obtain fully secure encryption of data it can be same that AES provides enough security for all data protection requirements. In recent times, people store a large variety of files, applications, photos etc. on cloud data storage. They conjointly share confidential data and images whereas communicating over the internet. In same data server, data are stored by several users. Thus, it can be one of the main targets for hackers. Therefore, enough security is required. It’s conjointly the fact that users need faster access to their data. Within the case of large files, it takes a long time. Therefore, faster encryption method with affordable security is required.
In this thesis, a modified version of the Advanced Encryption Standard (AES) is proposed. There are some changes here in this algorithm like key generation and replacing MixColumn by Transposition. It provides faster encryption or decryption process. It makes the calculation simple. Thus it's quite easy to implement in a practical situation. It’s appropriate for any public clouds or communication over networks wherever security, also as quicker access, is required for the users. It can also be used in personal computers for storing sensitive data. From the experimental result, it's shown that the modified version of AES reduces the quality of calculation, provides a simpler way of implementation and provides a high range of efficiency. Some improvements may be done in the longer term because we've to compromise some securities here. So in future, security can be increased with easier implementation by some modifications. Thus, this

algorithm will be a lot more efficient and can provide more security.

References

[1] A. Sachdev, M. Bhansali, “Enhancing Cloud Computing Security Using AES Algorithm,” International Journal of Computer Applications (0975-8887), April 2013.

[2] K.M. Akhil, M.P. Kumar and B.R. Pushpa, “Enhanced cloud data security using AES algorithm,” International Conference on Intelligent Computing and Control (I2C2). DOI:10.1109/i2c2.2017.8321820

[3] P. Priyanka, R. Rayarikar and S. Upadhyay, "Modifications to AES algorithm for complex encryption," IJCSNS International Journal of Computer Science and Network Security (2011): 183-186.

[4] S. P. Jadhav and B. R. Nandwalker, “Efficient Cloud Computing with Secure Data Storage Using AES,” International Journal of Advanced Research in Computer and Communication Engineering, June 2015.

[5] P. V. Nithyabharathi, T. Kowsayla and V. Baskar, “To Enhance Multimedia Security in Cloud Computing Environment Using RSA and AES,” International Journal of Science, Engineering and Technology Research (IJSTER), February, 2014.

[6] V. R. Pancholi and B. P. Patel, "Enhancement of cloud computing security with secure data storage using AES," International Journal for Innovative Research in Science and Technology 2.9 (2016): 18-21.

[7] R. Arora and A. Parashar, “Secure User Data in Cloud Computing using Encryption Algorithm,” International Journal of Engineering Research and Applications (IJERA) Vol.3, Issue 4, (2013) ISSN: 2248-9622.

[8] P. Kawle, A. Hawse, G. Bajde, E. Tekam and R. Kalbande, “Modified Advanced Encryption Standard,” International Journal of Soft Computing and Engineering (IJSCE), 4 (2014)

[9] C.C. Lu and S.Y. Tseng, “Integrated Design of AES (Advanced Encryption Standard) Encrypter and Decrypter,” Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP’02) 1063-6862/02

[10] D. S. A. Elminaam, H. M. A. Kader and M. M. Hadhoud, “Evaluating The Performance of Symmetric Encryption Algorithms,” International Journal of Network Security, Vol.10, No.3, PP.213–219, May 2010

[11] A. Moh’d, Y. Jaraweh and Lo’ai Tawalbech, “AES-512: 512-Bit Advanced Encryption Standard Algorithm Design and Evaluation,” 2011 7th International Conference on Information Assurance and Security (IAS)

[12] G. Bertoni, L. Breveglieri, P. Maistri and V. Piuri, “Error Analysis and Detection Procedures for a Hardware Implementation of the Advanced Encryption Standard,” IEEE TRANSACTIONS ON COMPUTERS, VOL. 52, NO. 4, APRIL 2003

[13] R. Hobson and S. Wakelin, “An Area-Efficient High-Speed AES S-Box Method,” Proceedings of the 9th International Database Engineering & Application Symposium (IDEAS’05)

[14] E. Thambiraja, “A Survey on Various Most Common Encryption Techniques,” vol. 2, no. 7, pp. 226–233, 2012

[15] C. Wang, Q. Wang, K. Ren, and W. Lou, “Ensuring Data Storage Security in Cloud Computing,” 2009.

[16] S. Gunasekaran, M. P. Lavanya, “A Review on Enhancing Data Security in Cloud Computing Using RSA And AES Algorithms,” International Journal of Advances in Engineering Research IJAER), Vol. No. 9, Issue No. IV, April 2015

[17] J. Yan, F. Chen, “An Improved AES Key Expansion Algorithm,” In International Conference on Electrical, Mechanical and Industrial Engineering (2016).

[18] W. Stallings,”Cryptography and Network Security: Principles and Practice,” 6th Ed. Pearson (2014).

Md. Shafiul Alam Forhad completed B.Sc. Engg. degree in Computer Science and Engineering from Chittagong University of Engineering & Technology (CUET) in 2014 with outstanding result. He is now pursuing his M.Sc. Engg. degree in Computer Science and Engineering from the same university. His current research concerns are Cryptography, Machine Learning, and Data Mining. He was a lecturer in the Department of Computer Science and Information Technology of Patuakhali Science & Technology University. Now, he has been serving as a faculty member in the Department of Computer Science & Engineering (CSE), Chittagong University of Engineering & Technology (CUET), Bangladesh.

Siam Riaz is pursuing his B.Sc. Engg. degree in Computer Science and Engineering from Chittagong University of Engineering & Technology (CUET). He is now looking forward to an opportunity for working in a challenging environment where he can utilize his knowledge for developing his career.

Md. Sabir Hossain received Bachelor degree in Computer Science and Engineering from Chittagong University of Engineering & Technology (2015) with outstanding result. He is now pursuing his M.Sc. degree in Computer Science and Engineering from the same university. His current research interests are Machine Learning, Data Mining, Big data, and Software Engineering. Now, he has been serving as a faculty member in the Department of Computer Science & Engineering (CSE), Chittagong University of Engineering & Technology (CUET), Bangladesh.

Mrinmoy Das completed B.Sc. Engg. degree in Computer Science and Engineering from Chittagong University of Engineering & Technology (CUET) in 2014 with an excellent result. He is now obtaining his M.Sc. Engg. degree in Computer Science and Engineering from Bangladesh University of Engineering and

Technology (BUET). His current research interests are Machine Learning, Security & Cryptography, Human Computer Interaction, and Data Mining. He is currently serving as a faculty member in the department of Computer Science and Engineering of Bangladesh University of Business and Technology (BUBT).