International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 2, Issue 9, September 2015

Double AES Framework Based on Mode Selection

[1]Kamalendu Behera [2] K. Sudhir Kumar
[1][2]Dept. of Computer Science & Engineering
College of Engineering & Technology
Bhubaneswar, India
[1]kamalendubehera.kb@gmail.com, [2]ksudhircse@gmail.com

Abstract: **Since World War 1, individuals as well as organization were really concerned regarding the security of their data. From that period till now cryptography [20] has been playing a critical role. A lot of researchers have been doing a lot of work regarding the improvement of the security levels. Till now a number of solutions are being implemented to secure the data. Various cryptographic algorithms like RC4[19], DES[13], Triple-DES, AES, RSA[18], DSA etc. have come into picture. But if we check the history of, when an algorithms is developed it looks like it cannot be broken, but some day someone does break it. Then again a new algorithm takes birth and the history is repeated again. Among these based on security levels and speed of encryption and decryption, AES[21] is considered by NIST and other researchers as most efficient. Though it is provides best security to data, there is no guaranty that it is unbreakable forever. AES (Advanced Encryption Standard) is a symmetric key cryptographic algorithm. Based on the length of the key there are 3 versions of AES, which are, AES-128, AES-192, and AES-256[20]. In this research we are proposing an efficient framework using these three versions of AES with different combinations to strengthen the cipher against the attackers especially brute force attacks and side channel attacks. The proposed system is tested with text as well as images and the results are significant. This framework's design is feasible, reliable as it provides high level security and is also time efficient, when compared to the processing time of the algorithms individually. This system can be used by cloud servers, banking organizations, message forwarding systems, defense organizations, e-commerce entities[25] etc.**

***Key Words*—Security, Cryptography, Advanced Encryption Standard, Brute Force Attack, Side-Channel Attacks.**

I. INTRODUCTION

Data those are highly confidential, like OTP (One Time Password), ATM pins, passwords, transaction details, financial records, accounting details, health data, cloud data, Government data, trade secrets, intelligence/research documents, defense data, confidential business or personal data needs protection when they are being used or sent over unsecured networks and even when they are not being used. Since 4000 years, techniques for securing data are being developed and few years" later attacks on those techniques are also found. Till now there is no technique invented which gives 100% security and is free from any attacks. So the primary approach taken by researchers and scientists is to develop systems that show improved results than existing systems considering the level of security and processing speeds.

Before even computers existed, the practice of encryption and decryption was practiced which was known as classic cryptanalysis. It started from ancient Egypt hieroglyphic inscriptions, then Greek Systole used by Spartans, later Caesar Cipher in Rome till Choctaw Code talkers in World War I and Enigma machine & Purple in World War II [26]. But now in modern times, the private key and the public key algorithms [20] are playing the role of modern cryptosystems.

Reviewing the history of all the private and public key algorithms, none can guarantee that the system is unbreakable forever. But more the time is required to break a system, then better is the algorithm

II. EXISTING SOLUTIONS

Since attackers continuously keep on trying to break the cryptosystems, new improved algorithms are developed. The primary goal while developing a cryptosystem is reducing the probability of success for an attacker.

A. Study of Methods Used To Improve Encryption Algorithms Robustness

Luminita Scripcariu[5] suggested few techniques of modifying existing algorithms of AES by using dynamic S-Box and using dynamic permutation functions defined on GFs. It was also suggested to use a larger data structure and some good standards to reduce the correlation between plaintext and ciphertext. These methods decrease the probability of successful attacks. But they also create little additional work for the authorized entities as they have to maintain a track of S-Box[28] and other functions if they are dynamic. There is a probable chance that inverse S-Box and invertible functions used during the decryption are not exactly corresponding to the S-Box and permutation functions used during encryption. In that case system also fails for authorized personnel.

B. Application of Hybrid Encryption Algorithm in Software Security

Wang et.al[8] proposed a system with combination of three different algorithms which are (1) Initial Encryption (2) Base64 Encryption (3) Improved Vigenere Encryption. These three are used in particular order to encrypt data and the reverse order was used to decrypt it. The idea of using multiple algorithm is beneficial however if there would have been a possibility of changing order then the strength of the hybrid system would be even higher significantly.

C. Design of a Double AES Processor

Uttar et.al[11] proposed a double AES system, which is considered as an extension of AES. In this system a data is encrypted and decrypted twice with 2 different keys. Since the encryption and decryption is done twice, time consumed is also doubled. But this time is acceptable as this system is gives an excellent challenge to the brute force attackers. If keeping the time constant the strength of algorithm is further enhanced then results will be highly significant.

III. PROPOSED SOLUTION

A. Selecting AES

With an idea of extending the performance of a cryptosystem, our first task was to choose the best algorithm till date and derive a new system which can further reduce the probability of attacks. Among the well-known cryptographic algorithms, the Advanced Encryption Standard (AES) was selected by US National Institute of Standards and Technology (NIST) from 15 competing algorithms. NIST officially adopted AES on 6th December 2001[10], focusing on its strength of performance on almost all platforms and due to its ease of hardware implementation[23].

AES is a symmetric key cryptographic algorithm[16], where the same key used for both encryption and decryption is kept secret. Based on the key length there are 3 versions of AES which are given in Table I. The intermediate steps involved in any AES version are AddRoundKey ( ); SubByte ( ); ShiftRows ( ); MixColumns ( ) are iteratively repeated. Block size of data used for any one of them is 4 that is 128 bits.

Table I: Respective Key Length Block Size & No of Iterative Rounds for AES Types

AES Types	Key Lengths	Block Size	Iterative Rounds
AES-128	128 bits	4	10
AES-192	192 bits	4	12
AES-256	256 bits	4	14

Now using the idea of Double AES, we developed a framework that involves all three versions of AES into a single system with multiple combinations of two different AES algorithms. Figure 1 shows how a Double AES system works.

B. Framework Details

It is clearly understood from Figure I that in case of double encryption/decryption, the data is first encrypted by one of the AES versions using a particular key (Key_1) to generate a cipher data, then the cipher data is again encrypted by using another key (Key_2) to get another cipher which will be the final cipher. Then in the reverse order the decryption algorithms are carried out. This method reduces the correlation between the final cipher and the original data, making it more difficult for the attacker to solve it. To further make it complex for the attackers and reduce the probability of their success we made various combinations of double AES, like (AES 128 + AES 256), (AES 256 + AES 192) etc. We added a feature of Mode selection in the system based on Table 2.

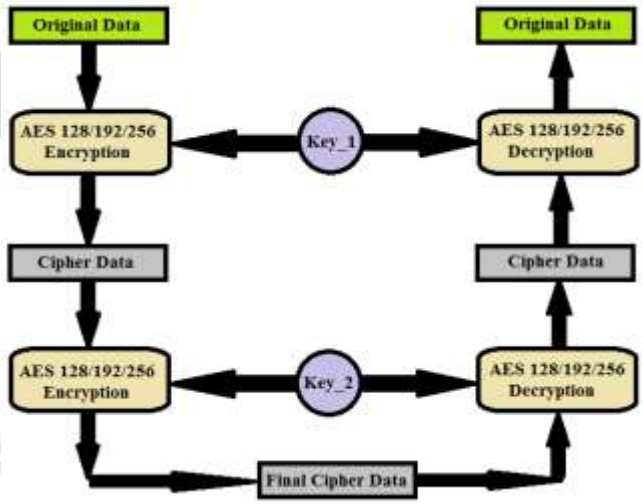


Figure I: Flow of Data in a Double Encryption System

We added a feature of Mode Selection to the system based on Table II, to strengthen it further. For a particular mode a particular combination of 2 algorithms with an order were chosen from the 3 given algorithms.

Table II: Various Combinations possible with Mode selection

Select Modes	AES 128	AES 192	AES 256
AES 128	Mode = 1	Mode = 2	Mode = 3
AES 192	Mode = 4	Mode = 5	Mode = 6
AES 256	Mode = 7	Mode = 8	Mode = 9

So in total there are 9 combinations possible, so user can manually select any mode from 1 to 9 which is again unknown to the attackers. Now with the mode selection feature added, the final framework is given in

Figure II. This mode selection will definitely increase the strength of double AES to 9 times.

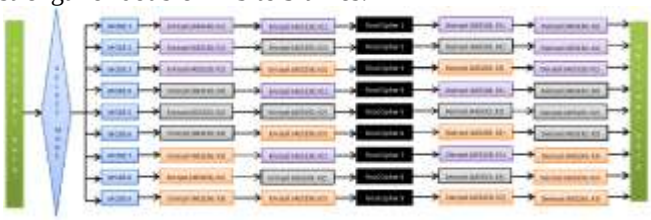


Figure II: Framework for the Proposed System

As seen in Table I key lengths for AES 128, AES 192 & AES 256 are different so let‟s consider the keys as K1 for AES 128, K2 for AES 192 & K3 for AES 256. So when a mode is selected, two keys are chosen respectively and a double AES processing id done as given in Figure 2. For example suppose Mode 6 is selected. The 1st key chosen is K2 and the original data is encrypted by AES 192 to generate a cipher. We can call it as intermediate cipher as this is not the final one. Then 2nd key is chosen as K3 and the intermediate cipher is encrypted again by AES 256 to generate another cipher. This is the final cipher. This final cipher is stored or sent over networks to the receivers. Now the decryption occurs in exact reverse order. 1st the final cipher is decrypted by AES 256 using K3 to generate the intermediate cipher and then that cipher is decrypted by AES 192 using K2 to generate the original data. All the modes function in the similar fashion.

IV. EXECUTION & RESULTS

This system is simulated in MATLAB R2012a (7.4.0.739) in a system with Intel ® Celeron processor, 1.5 GHz speed, having 2GB RAM & the OS is 64bit Windows 8. Following example is the encryption and decryption of few images that produces both intermediate as well as the final cipher image and decrypts it back to the original image. We have used modes arbitrarily.

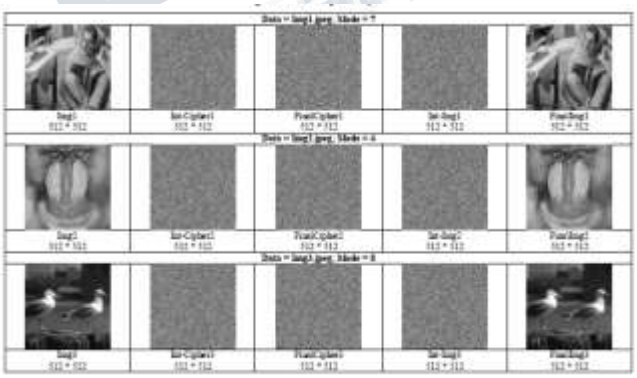
key128 = {„1f‟ „11‟ „aa‟ „22‟ „bb‟ „33‟ „cc‟ „44‟ „dd‟ „55‟ „ee‟ „66‟ „ff‟ „77‟ „88‟ „99‟}
key192 = {„01‟ „12‟ „23‟ „34‟ „45‟ „56‟ „67‟ „78‟ „89‟ „90‟ „ab‟ „bc‟ „cd‟ „de‟ „ef‟ „1a‟ „2b‟ „3c‟ „4d‟ „5e‟ „6f‟ „3a‟ „6b‟ „9c‟};
key256 = {„ff‟ „99‟ „f9‟ „9f‟ „dd‟ „88‟ „8d‟ „d8‟ „bb‟ „77‟ „b7‟ „7b‟ „ee‟ „66‟ „6e‟ „e6‟ „cc‟ „55‟ „5c‟ „c5‟ „aa‟ „44‟ „4a‟ „a4‟ „19‟ „91‟ „28‟ „82‟ „fa‟ „af‟ „cd‟ „dc‟};

Here we have used static keys as given above. But selection of key is user dependent they can either use constant key values for all data or they can use dynamic keys which are different keys for different data maintaining the specified key lengths. For testing the system we have selected 3 images Img1.jpeg, Img2.jpeg & Img3.jpeg. While processing Img1, we have selected the Mode as 7. So according to table II, the algorithms are AES 256 to be followed by AES 128 for encryption and the reverse order for decryption. Processing is done in following steps.

Step1: The image (Img1) is encrypted with AES 256 using key256 to generate an intermediate cipher (Int-Cipher1).
Step2: The Int-Cipher1 is again encrypted with AES 128 using key128 given above to generate the final cipher (FinalCipher1).
Step3: The FinalCipher1 is then decrypted with AES 128 using key128 which gives intermediate image (Int-Img1).
Step4: Finally the Int-Img1 is decrypted with AES 256 using key256 the get the final image (FinalImage1) which is the same as the original image (Img1).

Same 4 steps were repeated with other images Img2 & Img3 with mode 4 and 8 respectively, and the consolidated images & respective results of cipher generation are shown in Table III.

Note: The system works for characters also but the length of characters needs to be in multiples of 16, as AES algorithm is a block cipher algorithm, where the block size is 4. So it will divide the data into 4*4 matrices. In case when length is not in multiples of 16, the data can be padded with 0‟s to make the length a multiple of 16 before encryption and after decryption the added bits can be removed.



V. ADVANTAGES OF PROPOSED SYSTEM

Though the double encryption is expensive when execution time is compared with single encryption, the system has got various other worthy advantages when security levels are considered. It has additional strengths against 2 major known attacks.

A. Against Brute-Force Attacks

Since the data is encrypted twice the correlation between the cipher and original data. So double AES makes it remarkably strong and with modes, since there are 9 possible combinations of double AES that strength is again

multiplied to 9 times, hence creating a bigger search space for the brute-force attackers[6] making it difficult for them to guess the keys.

B. Against Side-Channel Attacks

Every time, when a different mode is selected, the flow of data, the execution times, acoustic (noise), fluctuations in electromagnetic waves and even the power consumption will be different. Side-Channel attackers[24][27]. track the data movements in & out from memory, note the execution times consumed by the system, track the fluctuations in radio-waves, heat & noise etc. to evaluate and determine the key. This system creates a bigger boundary for the side-channel attackers too.

VI. CONCLUSION

No doubt encrypting a data twice enhances the strength of the cipher by further reducing the correlation between the cipher and original data. By implementing the proposed system one can increase that enhanced strength to 9 times when brute force attack is considered. On the other hand since variable combinations of algorithms are being used it will also be difficult for the side channel attacker to track the details for timing attacks, power analysis attacks and acoustic attacks.

ACKNOWLEDGMENT

The authors are thankful to the anonymous reviewers for their comments and suggestions, which helped in improving the technical and editorial quality of this paper. Authors also acknowledge the authors who have published all related articles for providing both abstract and detailed ideas required for this paper. Last but not the least authors thank all of them have directly and indirectly been a part of this research and publication.

REFERENCES

[1]W. Stallings, 2010, “Cryptography and Network Security”.

2. M. D. Ashwini, S. D. Mangesh and N. K. Devendra, 10 April 2011 “FPGA Implementation of AES Encryption and Decryption”. pp. 401-405.

3. Chen-Hsing Wang, Chieh-Lin Chuang, and Cheng-Wen Wu, April 2010, “An Efficient Multimode Multiplier Supporting AES and Fundamental Operations of Public-Key Cryptosystems” pp. 553-563.

4. Alessandro Barenghia, Guido M. Bertonib, Luca Breveglieria & Gerardo Pelosia, 7th March 2013, “A fault induction technique
based on voltage underfeeding with application to attacks against AES and RSA”. pp. 1864-1878.

5. Luminita Scripcariu, 2015, “A study of methods used to improve encryption algorithms robustness”, pp. 1-4.

6. Abhishek Joshi, Mohammad Wazid, R. H. Goudarc, “An Efficient Cryptographic Scheme for Text Message Protection against Brute Force and Cryptanalytic Attacks”, pp. 360-366

7. Issam Hammad, Kamal El-Sankary and Ezz El-Masry, Sep. 2010 “High-Speed AES Encryptor With Efficient Merging Techniques”, pp. 67-71

8. Xinqiang Li; Lili Yu; Lihuan Wei, 2013, “Application of Hybrid Encryption Algorithm in Software Security”, pp. 762-765.

9. Kou-Min Chengy, Ting-Yi Changz, Jung-Wen Lo, “Cryptanalysis of Security Enhancement for a Modifed Authenticated Key Agreement Protocol”, pp. 1-5. Available at http://ijns.femto.com.tw/paper_upload/IJNS-2009-12-10-1.pdf

10. William E. (Bill) Burr, April 2003, “Selecting the Advanced Encryption Standard”.

11. Uttam Kumar Roy and Md. Liakot Ali, 22 December 2012, “Design of a Double AES Processor”, pp. 466-469.

12. Khelifi A, Aburrous M, Talib M.A., Shastry P.V.S., July 2013, “Enhancing Protection Techniques of E-Banking Security Services Using Open Source Cryptographic Algorithms”.

13. D. M. M. Alani, Jun. 2010, “DES96 - Improved DES Security”, IEEE 7th International Multi-Conference on Systems, Signals and Devices pp. 4244-7534.

14. F. J. Kherad, M. V. Malakooti, H. R. Naji and P. Haghighat, “A New Symmetric Cryptography Algorithm to Secure E-Commerce Transactions”, IEEE International Conference on Financial Theory and Engineering, 12-15 Mar. 2010, pp. 4244-7759.

15. Arundhati Joshi, P. K. Dakhole, Ajay Thatere, 20th March 2015,”Implementation of S-Box for AES”

16. F. Shao, Z. Chang and Y. Zhang, Apr. 2010, "AES Encryption Algorithm Based on the High Performance Computing of GPU", IEEE Second International Conference on Communication Software and Networks, pp. 7695-3961.

17. HAN Yu, ZOU Xue-cheng, LIU Zheng-lin, CHEN Yi-cheng, December 2008, "The research of DPA attacks against AES implementations", pp. 101–106.

18. Hsiao-Ying Lin & Wen-Guey Tzeng, June 2012, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding", pp. 995-1003

19. Sourav Sen Gupta, Anupam Chattopadhyay, Koushik Sinha, Subhamoy Maitra and Bhabani P. Sinha, April 2013, "High-Performance Hardware Implementation for RC4 Stream Cipher", IEEE Transactions On Computers, pp. 730-743

20. Marshall D. Abrams and Harold J. Podell, "Essay on Cryptography", pp. 350-385

21. NIST, January, 2010, "DRAFT NIST Special Publication 800-131, Recommendation for the Transitioning of Cryptographic Algorithms and Key Sizes", Federal Information Processing Standards Publication (FIPS PUB) 197, National Institute of Standards and Technology (NIST).

22. Mohanaraj Patchappen, Yaszrina Mohd. Yassin, Ettikan K. Karuppiah, 2015 "Batch Processing of Multi-Variant AES Cipher with GPU". pp. 32-36

23. G. Leopld, December 10, 2001, "U.S. unveils advanced encryption standard," EE Times. Available online at: http://www.eetimes.com/story/OEG20011205S0060.

24. Konrad J. Kulikowski, Mark G. Karpovsky, Alexander Taubin, 2007, "Robust codes and robust, fault-tolerant architectures of the Advanced Encryption Standard". pp. 139–149

25. Gilad Parann Nissany, 19th Nov2013, "Data that require Cloud Encryption", all categories of sensible data. Available online at http://www.porticor.com/2013/11/cloud-encryption-data/

26. Source Wikipedia "History of Cryptography". Available at https://en.wikipedia.org/wiki/History_of_cryptography

27. Deepa G.M., G. SriTeja & Professor S. Venkateswarlu, ISSN:2249-5789, "An Overview of Acoustic Side-Channel Attack", pp. 15-20

28. Edwin NC Mui Custom R & D Engineer, "Practical Implementation of Rijndael S-Box Using Combinational Logic"

ABOUT AUTHORS

Kamalendu Behera received the B.Tech degree in Information Technology from C. V. Raman College of Engineering, India in 2009 and worked with Infosys & Dell for 2 years & a part time teaching experience of 2 years. He received M.Tech degree from College of Engineering & Technology in 2015. He is interested in cloud computing, cryptography, number theory and image processing.

K. Sudhir Kumar received the Masters in Computer Application degree from Gayatri Institute of Computer & Management Studies in 2013. He received M.Tech degree from College of Engineering & Technology in 2015. He has a part time teaching experience of 5 years since 2010. His research interests include Soft Computing, Computer Architecture and Image Processing.