Extended AES Algorithm with Custom Encryption for Government-level Classified Messages

Sreyam Dasgupta, Pritish Das

Abstract*: The paper is primarily concerned with the data security issues faced while sending the data over the network. The issues are can be avoided with the proposed algorithm: Extended AES Algorithm with Custom Configurable Encryption. The added layer of security is based on the Caesar Cipher encryption algorithm. Although the algorithm is highly vulnerable to a few attacks, our modifications in the algorithm are tailor made to deny those attacks completely. The user has no idea that Caesar cipher is being used. Moreover, the key is changed for every word in the message, thus removing the vulnerability to frequency analysis attack. This layer will give some added protection to the underlying AES algorithm, which is already very secure. In today's electronic age, the importance of digital cryptography in securing electronic data transactions is unquestionable. Every day, users electronically generate and communicate a large volume of information with others. This information includes medical, financial and legal files; automatic and Internet banking; phone conversations, pay-per-view television and other e-commerce transactions as well as military information and some top-secret government intel. To meet these requirements, Advanced Encryption Standard (AES) for encryption of electronic data can be used. Governments prefer using AES for encryption of classified messages. Although no major attack on AES has been discovered yet, it is presumed that AES might have been broken without the attack being known to us. Thus, an added layer is used to make it safer.

***Index Terms*: AES, Cyber Security, Caesar Cipher**

I. INTRODUCTION

The most eccentric and widely used symmetric encryption algorithm most widely used at present is the Advanced Encryption Standard(AES). It is found at least six times faster than triple DES. Another solution for DES was needed as its key size was too small. With more computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback, but it was found to be sluggish.

Revised Manuscript Received on June 14, 2019

Sreyam Dasgupta, Computer Science, Vellore Institute of Technology, Kolkata, India.

Pritish Das, Computer Science, Vellore Institute of Technology, Jamshedpur, India.

AES is an iterative cipher rather than Feistel cipher. It is based on 'substitution-permutation network'. It has a series of linked operations, some of which replace inputs by specific outputs or Substitutions and others by shuffling bits which is also known as permutation.

AES does all its computations on bytes rather than bits. It uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key. For encryption it uses byte substitution, shiftrows, mixcolumns and addroundkey respectively. For decryption its the reverse of the encryption process.

Till date, AES has been widely adopted and supported both in hardware and software. No practical cryptanalytic attacks against AES have been found but security is only guaranteed only if it is correctly implemented and good key management is used. But as everything has some advantages and disadvantages, same case applies to AES as well. Cryptanalysts have found a flaw in AES that can crack secret keys faster than before. Key scheduling of AES – 192 and AES – 256 are weak. There have been attacks on these variants when used in a network. The cipher AES – 256 is used in SSL/TLS across the network. It's still the top cipher used in government organizations. In theory its not crack able since the combination of keys are massive. NSA recommends 128 bit key for encryption. AES – 256 is weaker than AES – 128. One should not use AES – 256 for building a hash function. Even a supercomputer would take 1 billion years to crack the 128 – bit AES key using brute force attack. This is more than the age of the universe(13.75 billion years).

Caesar Cipher is a mono-alphabetic cipher where each letter of plaintext is substituted by another letter to form the ciphertext. It is made possible because of shifting by some fixed number between 0 and 25. A number between 0 and 25 becomes the key of encryption.

It is not a secure cryptosystem because there are only 26 possible keys to guess. A cryptanalytic (hacker) can compute an exhaustive key search with even limited computing resources.

II. LIRERATURE SURVEY

Neenu Shaji et al. (2015) in their paper titled "Design of AES architecture with area and speed tradeoff" have discussed about work addresses. The area optimization of AES can be achieved by mapping the transformations to lower data path hardware and consequently using iterative loop architecture. The proposed design achieves tradeoff between the speed and area without using BRAM. The data path is not set to 128 bits and hardware components are either 8 or 32 bits. The whole design is developed with the help of the software Xilinx ISE and is synthesized with its tools. The simulation and synthesis is done targeting the Spartan3 device. The results obtained show that their proposed AES Architecture is suitable for use in resource constrained systems [7].

Rizky Riyaldhi et al. (2017) in their paper titled "Improvement of Advanced Encryption Standard Algorithm with Shift Row and S.box Modification Mapping in Mix Column" have talked about the improvement caused by the reduction of shift row circular process and S Box modification for Mix Column transformation. Their achieved a percentage improvement of 86.143% on encryption process and 13.085% on decryption process. However, the proposed implementation consumes larger memory space to store two modified S Box map and Array Shift Row map [5].

Karim Shahbazi et al. (2017) in their paper titled "Design and implementation of an ASIP-based cryptography processor for AES, IDEA, and MD5" have discussed about a new ASIP-based crypto processor for AES, IDEA and MD5 design. The instruction set consists of both general-purpose and specific instructions for the above crypto- graphic algorithms. Hence, a software developer can select the encryption method. The results are very promising. The maximum achieved frequency is 166.916 MHz. It is in comparison with some of the other modern designs. This design also has higher throughput than other ASIP- based crypto processors. Still, several optimizations can be done in terms of the instruction-set. More than one crypto algorithm can be implemented in one single ASIP processor. This will allow the users to select an encryption algorithm they desire [6].

B.Nageswara Rao et al. (2017) in their paper "Design of Modified AES Algorithm for Data Security" used the method that the number of rounds has been increased from 10 to 16 and a key of size 320 bits is used instead of 128,192 or 256 bits. Polybius square is used for key generation. The advantages are security is increased security and transmission occurs at a high speed. The drawback for the paper is the system can still be hacked by taking more computational time [1].

Ako Muhammad Abdullah (2017) in his paper "AES Algorithm to Encrypt and Decrypt Data" implemented 10 rounds of AES encryption is used with a variable key length of 128, 192 or 256 bits. The advantage is AES is one of the most efficient algorithms for cyber security and well supported on software and hardware. The drawback for the same is the system can be hacked by taking more computational time.

N Sivasankari et al. (2017) in their paper "Implementation of Area Efficient 128-bit Based AES Algorithm in FPGA" implemented an arbitrary key generator (TRNG) is used to generate the mystery key. 128 bit state is divided into 32 bits and then shift row operation is performed. Different matrices are used for mix column functions. The advantage is the implied strategy requires low asset use and has a high throughput of 38.65 GBPS. The drawback is implementation is designed only for AES-128. No additional security than general AES implementation [2].

Shamir Scheme and AES Algorithm implemented the key is split into different keys who are then encrypted and stored. Shamir's threshold scheme is used for management of keys. The advantage is if one key is lost then other keys are used to recover the original key. It is difficult to hack the data storage in cloud by using multiple key managers. A single point of failure should not affect the availability of data. The drawback is only focuses on the key rather than the encryption or decryption process.

Talari BhanuTeja et al. (2017) in their paper "Encryption And Decryption – Data Security For Cloud Computing – Using Aes Algorithm" implemented both RSA and AES algorithms are used for encryption and decryption. Secure uploading and downloading of files is proposed. The advantage is The system provides a spine structure to cloud storage frameworks where security issues are required to be decreased as much as possible. The drawback is Proposed system works only on text files and not on data like image, audio, video, etc [4].

A. Limitations of Existing Systems

AES is one of the most secured encryption processes. In theory, it is not crack-able. But it is suspected that it has been cracked already, without anyone knowing about it. That's why different algorithms are being implemented to make it more secure. One of the primary reasons of such suspicion is due to the length of the key not being long enough. AES uses too simple algebraic structure which is very hard to implement with software. Every block is always encrypted in the same way. AES in counter mode is complex enough to implement in software taking both performance and security into considerations. The way of handling different types of attacks, by AES is explained below.The traditional Caesar Cipher encryption is common and can be broken relatively easily.

A.1 Brute Force Attack

AES keys are of different size, the minimum length being 128 bits. It can provide 2^{128} possible keys. Using brute force attack on this domain of keys is going to prove highly ineffective. Even if the original key is found after searching half the key set, it will still require searching 2^{127} keys.

Brute-Force attack is very effective in breaking caesar cipher encryption. Only 25 possible keys are there. Each one of them can be tried and the key which leads to useful word meanings can be chosen. From there, all other messages can be broken using that key.

A.2 Mathematical Attack

Multiplicative inverse of a given number in the Galois finite field is calculated and is used for the generation of S – box substitution table in AES. This helps in stopping all types of linear and differential cyber attacks.

A.3 Timing Attack

AES is susceptible to timing attacks. Timing attacks involve implementation level attacks on the algorithms which do not run in a fixed time. The S – box substitutions in AES is dependent on the key and it can take variable time while implementation – this is ideal for timing attacks. S – box substitutions may be removed but it will slow down the AES cipher, which is also not desirable.

A.4. Frequency analysis

Frequency Analysis is very useful in breaking caesar cipher encryption. The frequency of letters in ciphertext is compared with the English letter frequencies.

III. PROPOSED WORK

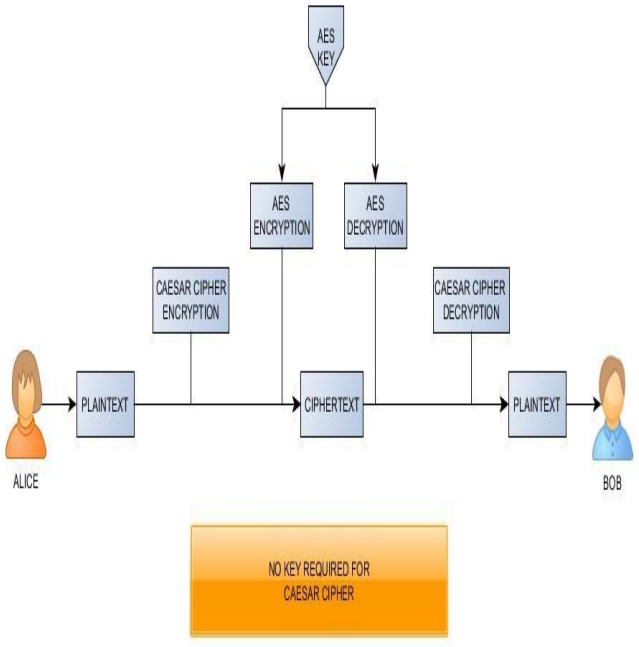

Fig – 1 Overview of the process

A. Proposed System Overview

The program begins with the sender, who wants to send a text message, known as known as plaintext in cyber security jargon, to the receiver. The plaintext is written in a textbox and when one hits the “encrypt” button the plain text is converted first to an intermediate ciphertext which is the result of encryption done by the Caesar cipher. The key generated for this encryption is the last letter of the plaintext and it is inserted at a particular index of the ciphertext depending on the length of the string being odd or even. At this stage the plaintext is converted to ciphertext but only 50%. The next step of encryption is AES encryption which uses standard AES algorithm to encrypt the intermediate ciphertext into a complete ciphertext which is very difficult to crack by the attacker.

At the receiver’s side, the decryption process begins. The decryption process is the reverse of the encryption algorithm. At first the ciphertext is decrypted using AES decryption and then further it is decrypted using the Caesar decryption, with the key being hidden in the cipher text only. After both the stages of decryption is completed the receiver will receive the plaintext as sent by the sender. The best part of this system is that no external key is required for the Caesar cipher to work as the key is chosen prior to the Caesar encryption as the last letter in the text.

The main reason for effectiveness of this process is that no outsider can guess the steps it is using, as nowhere it is specified that Caesar cipher is being used.

The user will land on a webpage which will ask 2 options:

ENCRYPT – This will encrypt the plaintext.

DECRYPT - This will decrypt the ciphertext.

Encrypt – Here the user will enter the plaintext in a textbox and on clicking ‘encrypt' the plaintext will be encrypted using custom made Caesar cipher encryption algorithm after which it will be further encrypted using AES Algorithm. Furthermore the user can custom configure it in order to make the ciphertext even more secure. To see the ciphertext of a particular plaintext one can select the required plaintext from the dropdown and can press ‘SHOW’ button to reveal the encrypted ciphertext. This is done on Alice’s or the Sender’s Page.

Decrypt – Here the user will be able to decrypt the ciphertext to plaintext using the reverse of AES algorithm with custom configuration of the Caesar Cipher decryption. This is done on Bob’s or the Receiver’s page.

This is done with the help of PHP programming language and writing the code for the AES algorithm.

After encrypting the plaintext to ciphertext, it goes to the database where it gets stored.

And for decrypting the same procedure is followed. The ciphertext is pulled from the database and converted back to plaintext.

Major use of this application will be in military and in various top-secret government intelligence agencies. Basically, any organization who needs to exchange messages with excessive security can implement this idea.

B. Algorithm

INPUT: Plaintext at Receiver's Side
OUTPUT: Plaintext at Sender's Side

BEGIN
1. RECEIVE PLAINTEXT
2. ENCRYPT USING MODIFIED CAESAR CIPHER
2.1 TAKE THE LAST LETTER OF THE PLAINTEXT
2.2 USING THE LAST LETTER AS KEY ENCRYPT THE PLAINTEXT USING CAESAR CIPHER
2.3 INSERT THE KEY IN THE MIDDILE OF THE INTERMEDIATE CIPHERTEXT
2.3.1 IF LENGTH OF PLAINTEXT IS EVEN PUT IT IN $((n / 2) + 1)^{th}$ POSITION
2.3.2 ELSE PUT IT IN $((n + 1) / 2)^{th}$ POSITION
3. ENCRYPT USING AES ALGORITHM TO GET FINAL CIPHERTEXT
4. SEND CIPHERTEXT AT SENDER'S SIDE
5. RECEIVE CIPHERTEXT AT RECEIVER'S SIDE
6. DECRYPT USING AES ALGORITHM TO GET INTERMEDIATE PLAINTEXT WITH KEY IN THE MIDDLE
7. DECRYPT USING MODIFIED CAESAR CIPHER
7.1 TAKE THE MIDDLE LETTER AS KEY
7.1.1 IF LENGTH OF CIPHERTEXT IS ODD THEN LETTER AT $((n + 1) / 2)^{th}$ POSITION IS THE KEY
7.1.2 ELSE LETTER AT $(n / 2)^{th}$ POSITION IS THE KEY
7.2 REMOVE THE KEY FROM THE MIDDLE OF THE TEXT
7.3 USE THE KEY TO DECRYPT THE CIPHERTEXT USING MODIFIED CAESAR CIPHER
8. DISPLAY PLAINTEXT
END

C. Time Complexity to Crack the Algorithm

The time complexity is determined with the assumption that the hacker knows that itis using modified Caesar Cipher along with AES encryption. This is not possible unless it is an inside attack. This is because, rarely Caesar ciphers are used as an added layer before AES encryption. Moreover,this algorithm is different than the traditional Caesar cipher. Different keys are used every time and it is based on last letter of each word of the input text. Hence, for a text of more than one-word, different keys are used.

For a message containing 5 words, 5 different keys can be used. This would require 5*O(n^2) complexity for the breaking of just the Caesar cipher layer. The best-known attack on AES encryption, where the hacker is denied user privileges, is of 2^32 complexity, till date. Thus, for a 5-word text, where a hacker knows about the layers of encryption, time complexity is 2^32*5*O(n^2).

D. Advantages over Existing Algorithm

Since this algorithm uses a different key for every different word, frequency analysis, which is a major way of breaking Caesar cipher, is not going to work. Counting the frequency of letters in each word to find the key for that word doesn't seem meaningful.

Another relatively easy way of breaking Caesar cipher is using brute-force by trying all the alphabets and seeing which letter makes sense. Now for one key it requires at most 25 tries. But when two keys are used it requires 25^2 tries for breaking the cipher. Hence for n different keys, it requires 25^n tries.

IV. RESULTS AND DISCUSSION

First we enter the plaintext. Here "gerrard" is entered as the plaintext.

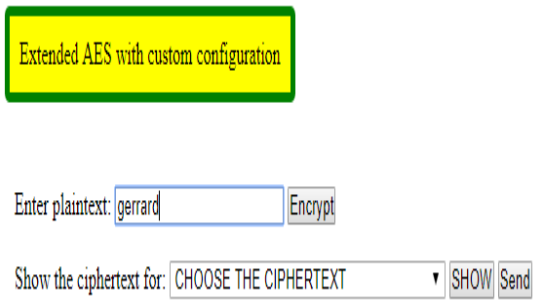

Fig -2: Snapshot of software where plaintext is taken as the input

Then after hitting the "encrypt" button we get the intermediate ciphertext using modified Caesar cipher encryption algorithm.

Final ciphertext is obtained using the intermediate ciphertext as plaintext for standard AES encryption algorithm.

Now decryption process begins and we get the final plaintext by decrypting the AES algorithm first and then the modified Caesar cipher.

Fig -3: Snapshot of software where we get the plaintext after the decryption process

A. Test Cases

TEST CASE-1	
Plaintext	Gerrard
Ciphertext1 (after Caesar Cipher encryption)	Kivvdevh
Ciphertext2 (after AES encryption)	aYvGLyUas5/8yyKmG2r5z Le1K9/BOUeswVej9x15urc=
Ciphertext3 (after AES decryption)	Kivvdevh
Ciphertext4 (after Caesar Cipher decryption)	Gerrard

Fig -4: Test Case 1

TEST CASE-2	
Plaintext	Ronaldo
Ciphertext1 (after Caesar Cipher encryption)	Gdcpoasd
Ciphertext2 (after AES encryption)	Z1oV0AQbdnHqJVfeZuC+Qg W+YFirbonrLyCpbBpsbWA=
Ciphertext3 (after AES decryption)	Gdcpoasd
Ciphertext4 (after Caesar Cipher decryption)	Ronaldo

Fig 5: Test Case 2

TEST CASE-3	
Plaintext	1234
Ciphertext1 (after Caesar Cipher encryption)	12434
Ciphertext2 (after AES encryption)	9VPVQWXlVhWxXBlABH0F Sq8gJHxrIlI2D4ErFV/1Wvk=
Ciphertext3 (after AES decryption)	12434
Ciphertext4 (after Caesar Cipher decryption)	1234

Fig -6: Test Case 3

B. Comparison with Other Algorithms

Blowfish is a symmetric key block cipher with key size of 32 - 448 bits and block size of 64 bits and which has about 16 rounds. The use of 64 bit block size makes it vulnerable to birthday attacks.

DES (Data Encryption Standard) is a symmetric key algorithm with key size of 56 bits + 8 parity bits and block size of 64 bits with 16 rounds. DES is insecure due to brute-force attacks. Other methods include differential cryptanalysis, linear cryptanalysis and improved Davies's attack.

Triple DES is a symmetric key algorithm with key size of 168, 112 or 56 bits and block size of 64 bits with 48 rounds. Practical Sweet32 attack is an example of attack on 3 - DES.

AES is a symmetric block cipher with key sizes 128, 192 or 256 bits and block size of 128 bits. It has 10, 12 and 14 rounds which depend on key size. The Biclique Attack and Related-key Attacks are some of the attacks which affect the AES algorithm.

ALGORITHM	KEYS	VULNERABILITY (KNOWN ATTACKS)	TIME COMPLEXITY TO CRACK THE ALGORITHM (or best known attacks)
BLOWFISH	32-448 BITS	BIRTHDAY ATTACK, DICTIONARY ATTACK	19 hrs
DES	56 BITS	BRUTE FORCE ATTACK	26 hrs
3-DES	3 DIFFERENT KEYS OF 56 BITS	BRUTE FORCE ATTACK, CHOSEN PLAINTEXT	25 mins
AES	128/ 192/ 256 BITS	SIDE-CHANNEL ATTACK	2^{32}, where hacker is denied user priviledges
OUR ALGORITHM	128 bits for AES, n different keys for n different words (max upto 26)	Same as AES	$5*O(n^2)*2^{32}$ (approximate)

Fig -7: Comparison with Other Algorithms

V. CONCLUSION

The paper is based on AES algorithm with Custom Configuration which is a completely new concept. A configurable algorithm is proposed that allows user to modify the algorithm each time he encrypts text, without the user actually knowing it. The algorithm uses AES and adds some custom configurable steps in the system. As is known the world is advancing more towards the online systems and internet accessibility worldwide is very good. Many governments use AES configuration for transmission of classified messages. Our algorithm is very useful for any such government, because it adds an extra layer of security that is completely unknown to people, even to the user. Thus, theoretically it cannot be broken without inside help. As is known the world is advancing more towards the online systems and internet accessibility worldwide is very good. Other systems can also use the proposed algorithm for message transmission since this is very highly secure. Future work may include making the system adaptable to alpha numeric inputs. In this paper, randomization of only the caesar cipher key is explained, but in the future, the first key for AES expansion can also be randomized based on the input.

REFERENCES

1. Rao B.Nageswara, Tejaswi, D., Varshini, K.Amrutha, Shankar, K.Phani, Prasanth B. "Design of Modified AES Algorithm for Data Security", International Journal For Technological Research In Engineering, Volume 4, Issue 8, pp 1289 – 1292 , April – 2017.
2. Sivasankari N, Rampriya K, Muthukumar, A, "Implementation of Area Efficient128-bit Based AES Algorithm in FPGA", European Journal of Advances in Engineering and Technology, 4(7), pp 541 - 548, 2017.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

3. Prof S. Athinarayanan, Priya, S. Nivetha,Supriya, R. (2017, Mar - Apr) "Secure Data with Key Managers by Using Shamir Scheme and AES Algorithm", International Journal of Computer Science Trends and Technology(IJCST), Volume 5, Issue 2, pp 298 - 301, Mar – Apr 2017.
4. Talari Bhanu Teja, Vootla Hemalatha,K Priyanka, "Encryption And Decryption – Data Security For Cloud Computing – Using Aes Algorithm", SSRG International Journal of Computer Trends and Technology (IJCTT) ,Special Issue , pp 80-83, April 2017.
5. Rizky Riyaldi, Rojali, Aditya Kurniawan, "Improvement of Advanced Encryption Standard Algorithm with Shift Row and S.Box Modification Mapping in Mix Column", 2nd International Conference on Computer Science and Computational Intelligence (ICCSCI), pp 401-407, 13-14 October 2017.
6. Karim Shahbazi, Mohammed Eshghi, Reza Faghih Mirzaee, "Design and implementation of an ASIP-based cryptography processor forAES, IDEA, and MD5", Engineering Science and Technology, an International Journal 20, pp 1308–1317, 2017.
7. NeenuShaji, Bonifus P. L, "Design of AES architecture with area and speed tradeoff", International Conferenceon Emerging Trends in Engineering, Science and Technology (ICETEST), pp1135-1140, 2015
8. T. Good and M. Benaissa, "AES on FPGA from the fastest to the smallest", UK Engineering and Physical Sciences Research Council (EPSRC), pp 1-14,2005

AUTHORS PROFILE

Sreyam Dasgupta Completed B-Tech from VIT University, Vellore. He is going to study Masters in Data Science at University of Glasgow. He has published two papers – "Image Compression using Bayesian Fourier" and "Smart Garbage Monitoring System" .

Pritish Das Completed B-Tech from VIT University. He is going to start working for Cerner Corporation.