Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 06, June-2022

Data Security using Advanced Encryption Standard(AES)

Snehal Kundlik Waybhase[1]
Department of Computer Science and Engineering,
G H Raisoni University,
Amravati, India

Prashant Adakane[2]
Asst. Prof., Department of Computer Science and Engineering,
G H Raisoni University, Amravati, India

***Abstract*— The daily demands of 60 percent of the world's population include the internet, which is a need in modern living. Data security and privacy are issues when using the internet because hackers can use this data. Therefore, taking this into consideration, we proposed the advance custom configurable algorithm for AES in this article. This is accomplished by adding an additional layer of protection to each letter of a message so that hackers cannot encrypt it. We are adding a new layer of encryption that is already specified by the algorithm, changing the keys for each letter and removing the vulnerability to frequent attacks. The already more secure AES algorithm is protected by this new layer. Digital encryption plays a crucial role in today's digital environment in protecting electronic data transfers. This information consists of documents related to finances and the law, to health, to automatic and online banking etc. The Advanced Encryption Standard (AES) for electrical data encryption can be utilized to meet these requirements. Although no significant AES attacks have been identified to far,**

Keywords:- AES(Advanced Encryption Standard); Internet; Security; Encryption layer;

I. INTRODUCTION

Advanced Encryption Standard, also known as AES, is widely used symmetric encryption technique. It is mostly utilized to encrypt and safeguard electronic data. Because it is faster and more effective than DES (Data encryption standard), it was used to replace DES. AES is a block cipher, and it comprises of three different block ciphers that are used to provide data encryption. Key sizes range from 128 to 256 bits. Data is encrypted in chunks of 128 bits.

Thus, from 128 bits of input, it produces 128 bits of encrypted cipher text as output. The substitution-permutation network principle, which underlies AES's functioning, involves a series of connected processes that replace and shuffle the incoming data.

How the cipher works :

Instead than operating on bits of data, AES operates on data in bytes. The encryption examines 128 bits (or 16 bytes) of the incoming data at a time since each block is 128 bits in size. According to the key length, the number of rounds will vary as follows:

- 10 rounds with a 128-bit key
- 12 cycles for a 192-bit key
- 14 cycles for a 256-bit key

Developing round keys :

All the round keys from the key are calculated using a Key Schedule algorithm. So the initial key is used to create many different round keys which will be used in the corresponding round of the encryption.

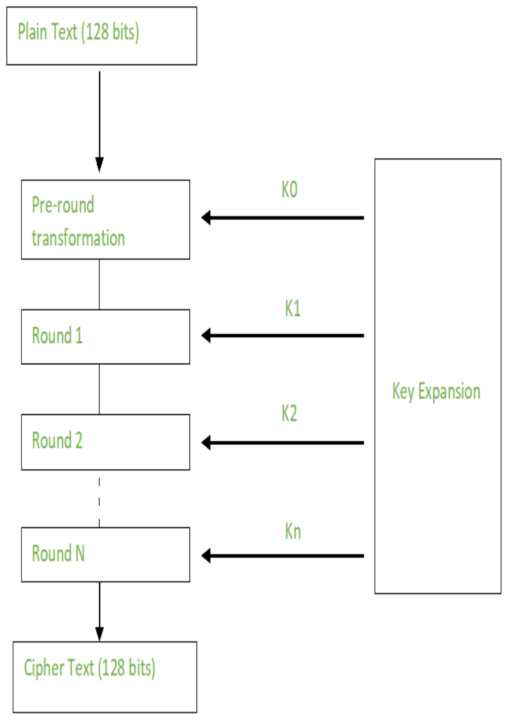

Fig.1 *Generating round keys process diagram*

II. LITERATURE SURVEY

NIST (National Institute of Standards and Technology) created AES in 1997. It was created to take the place of DES, which was slow and open to many attacks. In order to fix DES's flaws, a new encryption algorithm was developed. AES was subsequently released on November 26, 2001.

In their article "Design of Modified AES Algorithm for Data Security," B.Nageswara Rao et al. (2017) said that increasing the number of rounds (cycles) from 10 to 16 makes the algorithm (AES) more secure. As the number of cycles rises, more processing power will be needed, making it more challenging for hackers to break into the system. The key is created using the polybius square approach.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

In their 2017 paper titled "AES Algorithm to Encrypt and Decrypt Data," Ako Muhammad Abdullah implemented 10 rounds of AES encryption using keys that were 128 bits, 192 bits, and 256 bits in block cypher. His research leads to the conclusion that AES has greater security than competing algorithms like DES and 3DES.

N Sivasankari et al. (2017) report that both encryption and decryption have been actualized into a single chip (FPGA-XC5VLX50T), and that they perform with minimal resource utilisation and a high throughput of 38.65Gbps in their paper "Implementation of Area Efficient 128-bit Based AES Algorithm in FPGA."

Talari Bhanu Teja et al. (2017) implemented both RSA and AES technique are blended for encryption handle using USB device to upload and download data in their paper "Encryption and Decryption -Data Security for Cloud Computing -Using AES Algorithm." File uploading and downloads are archived securely. The system's advantage is that it gives cloud storage frameworks where security is concerned a spine structure. is elevated. The proposed system's limitation is that it only operates on text files, not other types of data like images. File uploading and downloads are archived securely. The system's advantage is that it gives cloud storage frameworks a spine structure to boost security. The flaw is that the proposed system only works with text files and not other types of data, such as image, audio, and video.

Because sensitive data is frequently transmitted and stored in today's digital technologies, encryption is widely used. The popular encryption algorithm used to secure data is AES, which is the industry standard. The problem of balancing area, power, and speed is difficult when designing VLSI systems, and hardware encryption is no exception. Certain performance factors are brought to the fore by system needs, but it is not always clear how to modify design implementations to satisfy performance requirements. There was a dearth of a single comparison analysis despite the fact that numerous resources in this field of study recognized and analysed interesting AES algorithm elements and their effects on a few of the design trade spaces.

The six AES elements that are addressed in this work are key size, mode specificity, round key storage, round unravelling, SBOX implementation, and pipelining. By looking at a compressed image of the resulting designs, readers may quickly analyse how each of the six aspects influences speed, power, area, latency, and throughput.

III. PRAPOSED WORK

Encryption: The sender, who wishes to send a text message to another user known as the recipient, initiates the application. The Caesar cypher is used to encrypt the plaintext, which is entered into a textbox and then converted to an intermediate ciphertext when the "encrypt" button is pressed. The last letter of the plaintext is used as the key in this encryption, and depending on whether the string is odd or even in length, it is inserted at a specific index in the ciphertext.

At this point, just 50% of the plaintext has been turned to ciphertext. The next stage of encryption is AES encryption, which turns the intermediate ciphertext into a complete ciphertext using the standard AES algorithm. The decryption procedure at the receiver's end starts with converting the ciphertext to plain text. the unlocking The decryption procedure is reversed in this phase.The ciphertext is initially decrypted using AES decryption, and then it is Caesar cypher decryption was used to open up the encrypted data; the key was only visible in the cypher text.Once the decryption process is finished, the recipient will receive the sender's original plaintext. Since Caesar cypher is not mentioned anywhere, no outsider can deduce the steps this technique uses, which is the major factor in its efficiency. A user-facing programme will present them with two options :

ENCRYPTION - Using a key, this will encrypt the plaintext to create the ciphertext.

DECRYPTION - Using the key, this decrypts the ciphertext to reveal the plaintext.

The textbox here is used to enter plaintext for encryption. When the user hits the "encrypt" button, the plaintext is first encrypted with the custom Caesar cypher encryption algorithm and subsequently with the AES method. Additionally, the user can modify it to further bolster the ciphertext's security. To reveal the encrypted ciphertext in its unencrypted form, select the required plaintext from the drop-down menu and then click "SHOW." At the Sender end, the user completes this activity.

Decryption: Using a customised Caesar Cipher decryption configuration and the AES approach in reverse, the user will be able to transform the ciphertext into plaintext in this case. On the receiver's end, this is done in the application.

This is accomplished by using the JAVA programming language and creating the AES algorithm's code. The plaintext is converted to ciphertext and then stored in the database. The same process is used for decryption as well, but in reverse. The ciphertext is taken out of the database and made plaintext again. The military and several top-secret government intelligence agencies will be the main users of this application. Basically, this concept can be used by any company that wants to exchange messages with high security.

IV. CHARACTERISTICS

- AES uses keys with lengths of 128, 192, and 256 bits.
- It is adaptable and has software and hardware implementations.
- It offers strong security and can stop numerous threats.
- Since it is copyright-free, anyone can use it anywhere in the world.
- For 128 bit keys, there are 10 processing rounds.

V. ADVANTAGES

- It offers consumers exceptional security and may be deployed on both hardware and software.
- It is one of the top open source options available for encryption.
- It is an extremely reliable algorithm.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

VI. RESULT

The text that i want to send, "Raisoni," must first be entered.

Raisoni Cipher text 1 in Plain Text (after Caser encryption) :
V/726+Q2jR+AxTKOUZ

Cipher text 2: wJkWGKIR8ODObFdc99nAGb6pOGU+1 DVUn7qADWVH4Dhp0 (after AES encryption)

Cipher text 3: V/726+Q2jR+AxTKOUZ (after AES decryption)

Cipher text 3 (after Caser decryption)

Raisoni

After clicking the "ENCRYPT" button, i will receive my cipher text and the key. I can now paste and copy.
The cipher text into another program and with the aid of another application, i can deliver the encrypted data to a different user.
For security purposes, i can even encrypt the key once more.
Now, I transmit the data in cipher text form to the other user.
I will receive the data into the receiver end.

VII. CONCLUSION

The extended AES algorithm with bespoke configuration, a brand-new idea, is the foundation of the work. The user can alter the algorithm each time text is encrypted with a programmable algorithm, without even realizing it. The technique incorporates some system-customizable phases and uses AES. As is well known, the world is moving more and more toward digital systems, and there is frequently decent internet accessibility everywhere. Many nations transmit classified messages using the AES setup. This technique gives an additional layer of protection that is fully hidden from everyone, including the user, making it incredibly beneficial for such governments as well as other organizations. Theoretically, it cannot be broken without assistance from inside.

VIII. REFERENCES

[1] Abdullah, A. M., & Aziz, R. H. H. (2016, June). New Approaches to Encrypt and Decrypt Data in Image using Cryptography and Steganography Algorithm., International Journal of Computer Applications, Vol. 143, No.4 (pp. 11-17).

[2] Singh, G. (2013). A study of encryption algorithms (RSA, DES, 3DES and AES) for information security. International Journal of Computer Applications, 67(19).

[3] Gaj, K., & Chodowiec, P. (2001, April). Fast implementation and fair comparison of the final candidates for Advanced Encryption Standard using Field Programmable Gate Arrays. In Cryptographers' Track at the RSA Conference (pp. 84-99). Springer Berlin Heidelberg.

[4] Stallings, W. (2006). Cryptography and network security: principles and practices. Pearson Education India.

[5] Yenuguvanilanka, J., & Elkeelany, O. (2008, April). Performance evaluation of hardware models of Advanced Encryption Standard (AES) algorithm. In Southeastcon, 2008. IEEE (pp. 222-225).

[6] Lu, C. C., & Tseng, S. Y. (2002). Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. In Application-Specific Systems, Architectures and Processors, 2002. Proceedings. The IEEE International Conference on (pp. 277-285).

[7] Mohamed, A. A., & Madian, A. H. (2010, December). A Modified Rijndael Algorithm and it's Implementation using FPGA. In Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on (pp. 335-338).

[8] Pramstaller, N., Gurkaynak, F. K., Haene, S., Kaeslin, H., Felber, N., & Fichtner, W. (2004, September). Towards an AES crypto-chip resistant to differential power analysis. In Solid-State Circuits Conference, 2004. ESSCIRC 2004. Proceeding of the 30th European IEEE (pp. 307-310).

[9] Deshpande, H. S., Karande, K. J., & Mulani, A. O. (2014, April). Efficient implementation of AES algorithm on FPGA. In Communications and Signal Processing (ICCSP), 2014 IEEE International Conference on (pp. 1895-1899).

[10] Nadeem, H (2006). A performance comparison of data encryption algorithms," IEEE Information and Communication Technologies, (pp. 84-89).

[11] Diaa, S., E, Hatem M. A. K., & Mohiy M. H. (2010, May) Evaluating the Performance of Symmetric Encryption Algorithms. International Journal of Network Security, Vol.10, No.3, (pp.213-219).

(This work is licensed under a Creative Commons Attribution 4.0 International License.)