IJIEMR Transactions, online available on 13th Apr 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 04

10.48047/IJIEMR/V12/ISSUE 04/131

Title FILE ENCRYPTION USING ADVANCED ENCRYPTION STANDARD (AES)

Volume 12, ISSUE 04, Pages: 1021-1029

Paper Authors
Mr. Ram Bhupal, Kollapudi Hemanth, Kamepalli Mahesh Babu , Kavuri Saatyakeya

FILE ENCRYPTION USING ADVANCED ENCRYPTION STANDARD (AES)

Mr. Ram Bhupal(Ph.D.)[1], **Assistant Professor**, Department of Information Technology, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur(Dt), Andhra Pradesh.

Kollapudi Hemanth[2], **Kamepalli Mahesh Babu** [3], **Kavuri Saatyakeya**[4]
[2,3,4,] Undergraduate Students, Department of Information Technology,
Vasireddy Venkatadri Institute of Technology, Nambur, Guntur (Dt), Andhra Pradesh.
[1] **hemanth8981@gmail.com , [2] mkamepalli02@gmail.com**
[3] **kavuri.saatyakeya05@gmail.com**

Abstract

File Crypt Manager is a desktop tool that can encrypt and decode any file, regardless of file format or size. Data stored on the computer should be kept secure against unauthorised access. Consider the consequences if the file includes private information that has been kept on the computer and is later hijacked by hackers. One method of data protection is to use cryptography to accomplish data encryption and decryption. For encrypting and decrypting information, this study used Advanced Encryption Standard (AES) symmetric key cryptography. With symmetric key encryption, both the encryption and decryption of files are performed using the same key. This project will allow users to choose files from their PCs and encrypt/decrypt them using a password.

Keywords: Advanced Encryption Standard (AES), Symmetric key Encryption, File Crypt Manager, Cryptography, Private information, Encrypt, Decrypt, Password

Introduction

Information security is extremely important in today's environment. Our digital infrastructure is more susceptible to cyberattacks as a result of the growing volume of data we store and communicate. Protecting the data from illegal access and usage is crucial. Encryption is among the greatest methods for accomplishing this. Data is transformed into a code using the encryption process, which is only accessible to those with the proper authorization. One of the most popular encryption methods will be covered in this article: Advanced Encryption Standard (AES) [1]

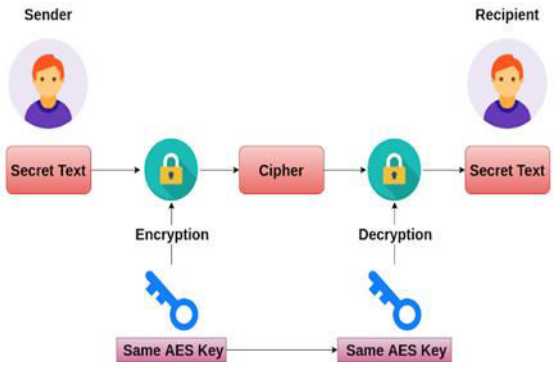
File encryption is the process of putting a file into an unreadable format that can only be viewed with the proper decryption key. This procedure makes sure that confidential information is safeguarded from unwanted access and can only be viewed by authorised users. Security of sensitive data, such as that found in medical records, financial records, and personal information, requires file encryption [2].

The Advanced Encryption Standard, a symmetric encryption technique, encrypts and decrypts data using a secret key. (AES). The AES method, developed by two Belgian cryptographers named Joan Daemen and Vincent Rijmen, was chosen as the default encryption algorithm by the National Institute of Standards and Technology (NIST) in 2001. In order to secure sensitive data, the US government approved AES, a symmetric encryption method, as a standard in 2001. It makes use of a block cipher, which encrypts data in distinct blocks of a particular size. [3,4]

Before we dive into AES, we need to comprehend the idea of symmetric encryption before we can fully understand AES. The term "symmetric encryption" refers to a method where both encryption and decryption are performed using the same key. This implies that both the person encrypting the data and the person decrypting it have access to the same key. The key for symmetric encryption needs to be kept safe. If an unauthorised person obtains the key, they can decrypt the data and get access to the information. Three phases make up the encryption and decryption process in symmetric encryption: key creation, encryption, and decryption [5,6].

Key Generation is the initial stage of symmetric encryption. To encrypt and decode the data, a random string of bits is utilised as the key. The strength of the encryption depends on the key's length. The strength of the encryption increases with the length of the key [7].

Data is encrypted using the key when it has been produced. The data is broken up into blocks, and each block is encrypted using the key. During the encryption process, data is jumbled, making it unreadable to anybody without the required key. When access is required, the data is decrypted using the same key that was used to encrypt it. The encryption procedure is reversed during decryption [8].



The most typical block size of AES is 128 bits, however for even better encryption and Decryption, AES may be set up to utilise 192 or 256-bit blocks. In order to protect sensitive data, governments, financial institutions, and other organisations utilise Currently used encryption techniques include the highly secure AES algorithm [9].

The security of AES is also aided by the key length employed. There are 2^128 potential keys when using the AES-128 algorithm since its key length is 128 bits. With a key length of 192 bits used by AES-192, there are 2^192 potential keys. There are 2256 keys that might be used using the AES-256 algorithm since the key length is 256 bits [10,11].

AES works by substituting and permuting data in a specific way. It uses a combination of substitution and permutation techniques to scramble the data[12]. This process is called a round, and AES can have up to 14 rounds, depending on the key length [13]. Each round consists of four steps:

SubBytes: In this step, the input bytes are replaced by corresponding values from the S-box. The S-box is a 16x16 lookup table that maps each byte value to a unique 8-bit value [14].

ShiftRows: The input matrix's rows are cycled through in this stage. The first row doesn't change, the second row is moved to the left by one position, the third row is moved by two places, and the fourth row is moved by three positions to the left. [15].

MixColumns: Each column of the input matrix is multiplied by a fixed matrix in this stage. In order to produce diffusion in the data, where each bit of the output depends on every bit of the input, this is done.

AddRoundKey: In this stage, every byte of the input matrix is XORed with its matching byte of the round key. Using a key schedule, the original key is used to produce the round key. [16].

The encrypted data is what was produced after the last round. The round keys are applied in reverse order, and the same processes are used to do decryption[17,18].

Literature Survey:

A popular technique for protecting sensitive data is file encryption using the Advanced Encryption Standard (AES). According to the research findings covered in this literature review, AES encryption for file encryption is quick, effective, and secure, with a variety of modes and key lengths available for use in a variety of applications. Key management is also essential for ensuring the security of encrypted files. Further research can focus on improving the efficiency and security of AES encryption for file encryption in different scenarios.

1."A Review on File Encryption Techniques Using AES Algorithm" by M. Kamatchi and S. Selvamuthukumaran

This study reviews various file encryption techniques that use AES algorithm, including the Cipher Block Chaining - CBC mode and the Counter - CTR mode. The authors also discuss the performance analysis of these encryption techniques and conclude that the CTR mode is more efficient than the CBC mode in terms of speed and security.

2."A Secure File Encryption System Using AES Algorithm" by S. Patel

This study proposes a secure file encryption system using AES algorithm, which can be used in various applications such as cloud computing, data storage, and data transmission. To increase security, the authors utilise a 256-bit key and the CBC mode of the AES algorithm. The suggested system's performance evaluation demonstrates its effectiveness and security.

3."A Comparative Study of File Encryption Techniques Using AES Algorithm" by N. Devi and M. Kavitha

This study compares various file encryption techniques using AES algorithm, including the ECB (Electronic Code Book) mode, the CBC mode, and the OFB (Output Feedback) mode. The authors evaluate these encryption techniques based on security, speed, and memory consumption. The results show that the CBC mode is the most secure and efficient encryption technique among the three modes.

Problem Identification

Using a coding that can only be cracked with the right password or decryption key, file encryption software is used to safeguard critical data. Data should have a high level of security and secrecy, especially when it is transported or kept in an unsafe environment. This is the goal of file encryption software.

Software for File Encryption is Required

File encryption software is now more important than ever due to the growth in digital data transit and storage. Some of the main justifications for why file encryption software is required include the following:

1. Confidentiality: To protect the secrecy of sensitive data, file encryption software is necessary. Data is encrypted via software, making it accessible only with a password or decryption key. This guarantees that the data can only be accessed by authorised individuals.

2. Protection from Cyber Attacks: Software that encrypts files offers defence against cyberattacks like hacking and data leaks. It is more difficult for attackers to steal or misuse sensitive information when data is encrypted since it is unreadable to unauthorised parties.

3. Compliance with Data Protection Regulations: Organizations are required to secure sensitive data under data protection laws in many nations. One of the most important approaches to guarantee data security and organisational compliance with data protection laws is encryption.

4. Secure Data Sharing: File encryption software makes it possible to share data in a secure manner, especially when doing so over public networks like the internet. Secure data exchange is made possible by encrypted data, which cannot be intercepted or read by unauthorised parties.

Methodology

Key expansion, the initial round, and the main rounds are the three main parts of the AES encryption process.

Key Expansion

The original secret key is used in the key expansion procedure to create a set of round keys. The quantity of round keys required depends on the size of the key and the number of encryption rounds. For instance, if the key size is 128 bits and there are ten rounds in the encryption process, ten round keys are generated.

Initial Round

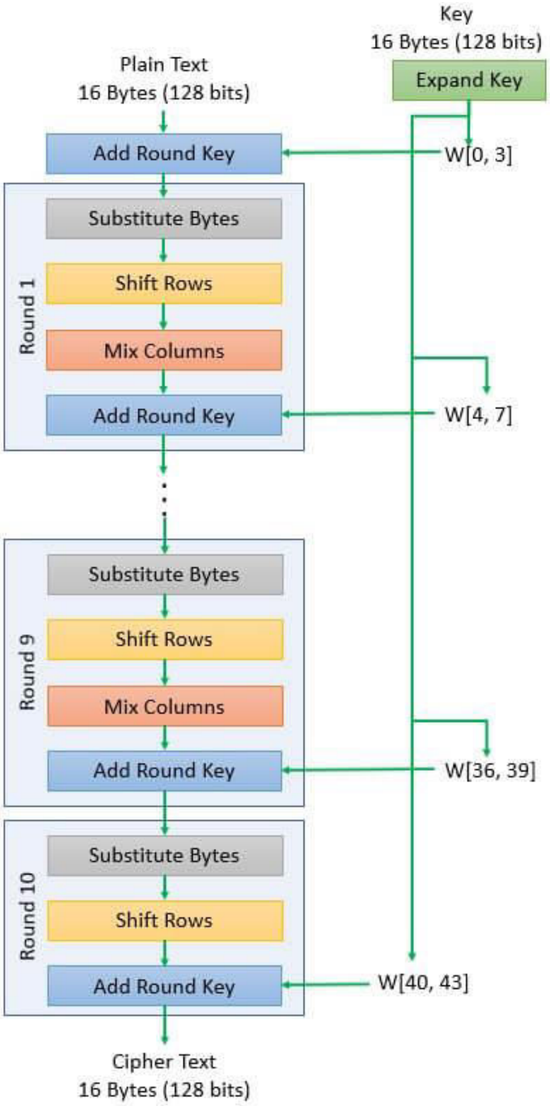
The plaintext data is XOR with the initial round key in the opening round. To create a new set of bits that are subsequently transmitted via the substitution box (S-box) and permutation box, the plaintext's bits and the round key are combined in the XOR operation (P-box).

Each byte of the input is replaced with a new byte via the non-linear substitution function known as S-box. The modular arithmetic, logarithms, and exponentiation operations are used to build the S-box.

The input data's bits are randomly shuffled by the linear permutation function known as P-box. The P-box is made to make it challenging to use statistical analysis to comprehend the output data. The output data from the P-box is intended to be difficult to interpret using statistical analysis.

Main Rounds

For each round key, the primary rounds include continually iterating the same set of actions. They include the mix-column operation, sending the outcome via the S-box and P-box, and XORing the output of the prior round with the key of the current round. The matrix multiplication technique known as mix-column offers further dispersion of the input data. It entails multiplying each input column by a predetermined matrix and then adding the results.



Depending on the size of the key and the overall number of encryption cycles, the main rounds are repeated for a certain number of rounds. The ciphertext, or encrypted form of the plaintext, is the result of the last major round.

Implementation

Encrypting files using AES encryption is a simple process that can be done using various software tools. Here is a step-by-step guide to encrypt files using AES encryption:

Step 1: Open an AES encryption application

There are software tools available that can encrypt files using AES encryption

Step 2: Choose the file to be encrypted

Choose the file to be encrypted from the device.

NOTE: You cannot encrypt the entire folder. But you can encrypt after archiving the folder like .zip .7z.tar and etc..

Step 3: Set a strong password

Set a strong password for the encrypted container. This password should be long, complex, and difficult to guess. It is the key that will be used to encrypt and decrypt so, remember that the key size determines the level of security and the speed of encryption and decryption.

Step 4: Click on the encrypt

Now select ENCRYPT option to encrypt the file. A new encrypted file with ".cryp" extension will be created in the same directory where the actual file is existed

Step 5: Access the encrypted files

To access the encrypted files, now select the file with the ".cryp" extension which you previously encrypted and enter the Secret Key during the time of Encryption. select DECRYPT option to decrypt the file. The decrypted file will be of the same name as before with the suffix "_decrypted_".

Step 6: Reset

Click on RESET option to reset the input fields and status bar.

Step 7: To Halt/Cancel

To halt the process by clicking on CANCEL option during Encryption/Decryption.

Results&Conclusion

Results

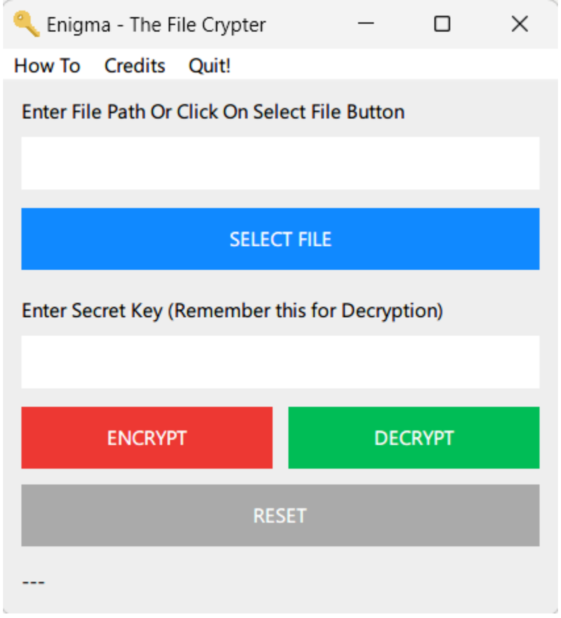
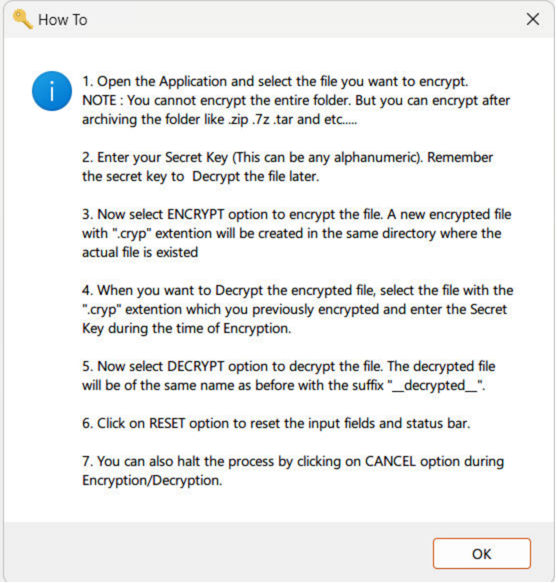

Enigma - The File Crypter
How To
Credits
Quit!
Enter File Path Or Click On Select File Button
SELECT FILE
Enter Secret Key (Remember this for Decryption)
ENCRYPT
DECRYPT
RESET

Figure1: Shows GUI of the screen before implementation.



1. Open the Application and select the file you want to encrypt.
NOTE : You cannot encrypt the entire folder. But you can encrypt after archiving the folder like .zip .7z .tar and etc.....

2. Enter your Secret Key (This can be any alphanumeric). Remember the secret key to Decrypt the file later.

3. Now select ENCRYPT option to encrypt the file. A new encrypted file with ".cryp" extention will be created in the same directory where the actual file is existed

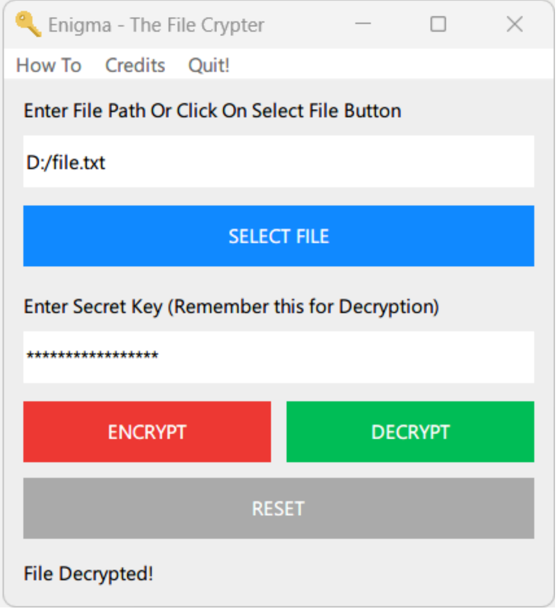
4. When you want to Decrypt the encrypted file, select the file with the ".cryp" extention which you previously encrypted and enter the Secret Key during the time of Encryption.

5. Now select DECRYPT option to decrypt the file. The decrypted file will be of the same name as before with the suffix "_decrypted_".

6. Click on RESET option to reset the input fields and status bar.

7. You can also halt the process by clicking on CANCEL option during Encryption/Decryption.

Figure2: A set of steps is displayed when 'How To' button is clicked for guidance.



Enter File Path Or Click On Select File Button

D:/file.txt

Enter Secret Key (Remember this for Decryption)

****

File Decrypted!

Figure3: When the file is selected from the machine and the password is set.

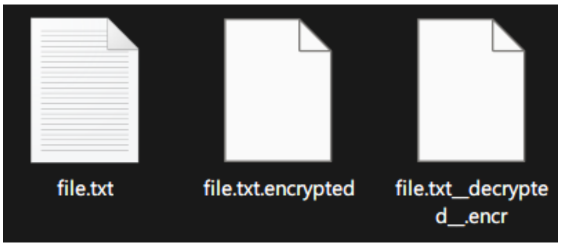
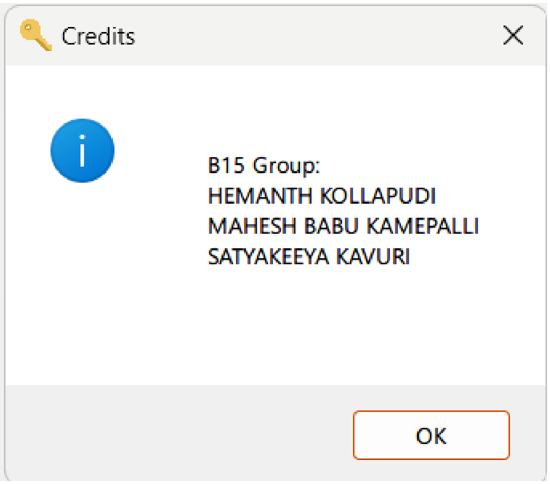


Figure4: The file selected for encryption is file.txt, file.txt.encrypted is formed after encryption, file.txt_decrypted_.encr is formed by decryption.



B15 Group:
HEMANTH KOLLAPUDI
MAHESH BABU KAMEPALLI
SATYAKEEYA KAVURI

Figure5: Credit for development of application

Conclusion

For file encryption, AES is a popular and very efficient encryption technique. High-level security, adaptability, and speed are all features it offers, making it the perfect option for several applications. AES encryption is susceptible to side-channel attacks and key management problems, among other flaws. To ensure the protection of sensitive data, AES encryption must be properly planned and put into practise. In general, AES encryption is a good approach to safeguard sensitive data and uphold data privacy.

The symmetric key encryption algorithm AES encrypts data in blocks of a predetermined size. AES encryption employs a block size of 128, 192, or 256 bits and a key length of up to 128, 192, or 256 bits. AES encryption works by dividing the file into fixed-size blocks, which are then individually encrypted with the key.

Several operating systems and software programmes employ AES encryption by default to safeguard data, and it is extensively used in a wide

Limitations & Future Work

Vulnerable to Side-Channel Attacks:

AES is susceptible to timing attacks and power analysis attacks, among other side-channel attacks. These attacks exploit vulnerabilities in the hardware or software used to implement AES, rather than the encryption algorithm itself.

Key management: The safety of the encryption key is a determining factor in the security of AES encryption. The encrypted data cannot be unlocked in the event of key loss or theft. AES encryption requires careful planning and execution for key management, which is a crucial component.

Limited Authentication: Authentication makes ensuring that the data hasn't been tampered with or altered while in route. Message authentication code (MAC) or a digital signature must be used in combination with AES encryption to offer authentication.

References

[1] Chen, L., & Sun, X. (2018). Research on File Encryption Algorithm Based on AES. Journal of Physics: Conference Series, 1065(5), 052008.

[2] Zou, L., Wang, X., & Huang, Q. (2018). File encryption scheme based on AES and RSA algorithms. Journal of Physics: Conference Series, 1029(1), 012032.

[3] Li, X., Li, Y., & Guo, X. (2015). A File Encryption Algorithm Based on AES and a New Chaotic Map. International Journal of Multimedia and Ubiquitous Engineering, 10(7), 203-212.

[4] Rana, R. S., & Kamboj, S. (2018). Comparative Analysis of Symmetric Key Encryption Algorithms. International Journal of Recent Technology and Engineering, 7(4), 337-342.

[5] Kim, Y. H., & Chung, Y. C. (2014). Secure file encryption using chaotic maps and AES. International Journal of Security and Its Applications, 8(4), 239-250.

[6] Lee, J. H., Kim, J. H., & Kim, K. J. (2016). A secure file encryption method based on AES and elliptic curve cryptography. Cluster Computing, 19(1), 97-106.

[7] Bhattacharya, S., & Sahoo, A. K. (2016). A comparative study of file encryption algorithms. International Journal of Computer Science and Network Security, 16(11), 44-51.

[8] Wang, Y., & Liu, X. (2016). A Secure File Encryption System Based on AES and Homomorphic Encryption. International Journal of Computer Science and Network Security, 16(2), 110-116.

[9] Shukla, M. K., & Mishra, S. K. (2018). Improved File Encryption Algorithm using AES and MD5. International Journal of Engineering and Technology, 7(2), 37-40.

[10] Zhou, W., Wang, X., & Qin, Z. (2014). A New File Encryption Algorithm Based on AES. Applied Mechanics and Materials, 606, 284-288.

[11] Chen, H., Hu, S., & Hu, Y. (2016). A New Hybrid Encryption Scheme based on RSA and AES for Secure Data Communication. Journal of Communications, 11(6), 536-543.

[12] Daemen, J., & Rijmen, V. (2002). The Design of Rijndael: AES-The Advanced Encryption Standard. Springer Science & Business Media.Engineering in medicine

[13] Nazir M, Wahid F, Ali Khan S. A Simple and Intelligent Approach for Brain MRI Classification. J Intell Fuzzy Syst 2015;28:1127–35. doi:10.3233/IFS-141396.

[14] Khurana, S., & Sharma, M. (2014). Performance evaluation of AES and DES encryption algorithms. International Journal of Advanced Research in Computer Science and Software Engineering, 4(6), 784-788.

[15] Kim, Y. H., Chung, Y. C., & Kim, J. W. (2015). A secure file encryption system using AES and SHA-256 for image data in the cloud. Journal of Supercomputing, 71(3), 1039-1053.

[16] Singh, N., & Kumar, P. (2015). Performance analysis of AES and Blowfish algorithms for file encryption. International Journal of Computer Applications, 121(19), 12-17.

[17] Huang, Q., Li, D., Li, L., & Xu, H. (2017). An improved file encryption scheme based on AES algorithm. Journal of Ambient Intelligence and Humanized Computing, 8(1), 95-102.

[18] Bishnoi, S. K., Singh, S., & Sharma, V. (2015). Implementation of File Encryption using AES Algorithm in Cloud Computing. International Journal of Engineering and Computer Science, 4(11), 15034-15038.