

Using AES Algorithm Encryption and Decryption of Text File, Image and Audio in Openssl and Time Calculation for Execution

Nikhil Anand

(School of Electronics Engineering, VIT University, Chennai, India)

Abstract: Cryptography is used to secure and protect data during communication. It is helpful to prevent unauthorized person or group of users from accessing any confidential data. Encryption and decryption are the two essential functionalities of cryptography [4]. A message sent over the network is transformed into an unrecognizable encrypted message known as data encryption. At the receiving end, the received message is converted to its original form known as decryption [2]. Encryption is a process which transforms the original information into an unrecognizable form. This new form of the message is entirely different from the original message. That's why a hacker is not able to read the data as senders use an encryption algorithm. Encryption is usually done using key algorithms. Decryption is a process of converting encoded/encrypted data in a form that is readable and understood by a human or a computer. This method is performed by un-encrypting the text manually or by using keys used to encrypt the original data.

Key Word: Cryptography; AES algorithm; Encryption and Decryption; Openssl tool; Security; Integrity.

Date of Submission: 10-12-2020

Date of Acceptance: 25-12-2020

I. Introduction

The Advanced Encryption Standard (AES) is a symmetric block cipher chosen by the U.S. government to protect classified information. AES is implemented in software and hardware throughout the world to encrypt sensitive data. It is essential for government computer security, cyber security and electronic data protection. The National Institute of Standards and Technology (NIST) started development of AES in 1997 when it announced the need for an alternative to the Data Encryption Standard (DES), which was starting to become vulnerable to brute-force attacks. NIST stated that the newer, advanced encryption algorithm would be unclassified and must be "capable of protecting sensitive government information well into the 21st century." It was intended to be easy to implement in hardware and software, as well as in restricted environments -- such as a smart card -- and offer decent defenses against various attack techniques. AES was created for the U.S. government with additional voluntary, free use in public or private, commercial or noncommercial programs that provide encryption services. However, nongovernmental organizations choosing to use AES are subject to limitations created by U.S. export control.

II. Methods

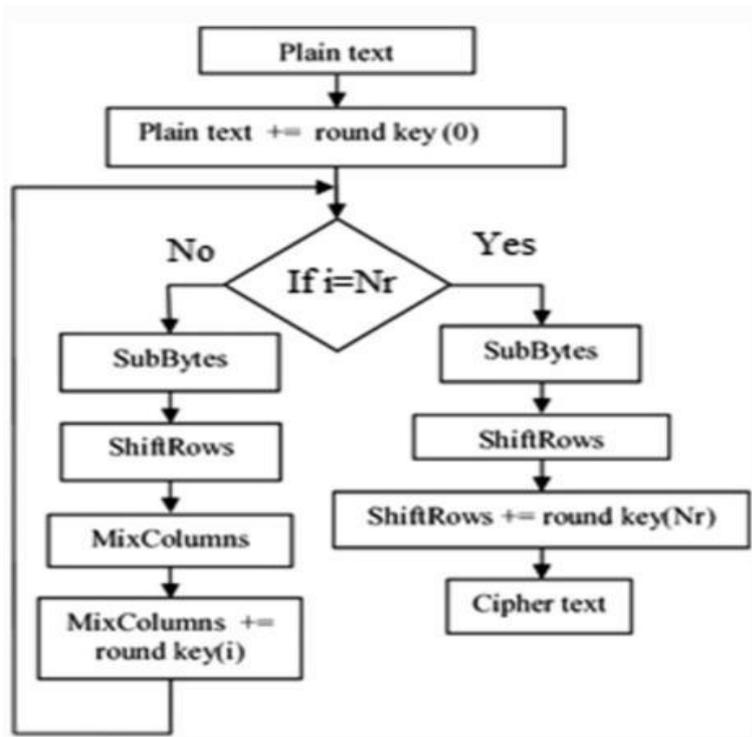
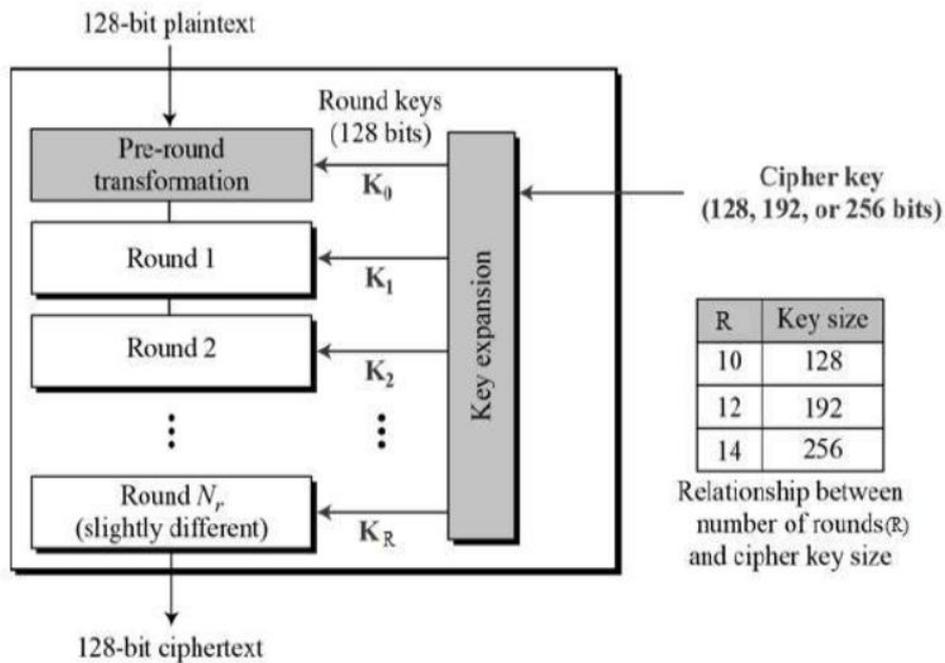
How AES encryption works

AES includes three block ciphers: AES-128, AES-192 and AES-256.

AES-128 uses a 128-bit key length to encrypt and decrypt a block of messages, while AES-192 uses a 192-bit key length and AES-256 a 256-bit key length to encrypt and decrypt messages. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128, 192 and 256 bits, respectively.

Symmetric, also known as *secret key*, ciphers use the same key for encrypting and decrypting, so the sender and the receiver must both know -- and use -- the same secret key. The government classifies information in three categories: Confidential, Secret or Top Secret. All key lengths can be used to protect the Confidential and Secret level. Top Secret information requires either 192- or 256-bit key lengths.

There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. A round consists of several processing steps that include substitution, transposition and mixing of the input plaintext to transform it into the final output of cipher text.



OPENSSL

Openssl is a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. It is also a general-purpose cryptography library. For more information about the team and community around the project, or to start making your own contributions, start with the community page.

Encryption of text file

Commands:

time openssl aes -256-cbc -a -iter 1024 -in index.php -out encrypt.enc : This command is used for encrypting the given file.

time openssl aes -256-cbc -a -d -iter 1024 -in encrypt.enc -out decrypt.decrypt : This command is used to decrypt the encrypted file

Encryption and decryption of images

Commands :

time openssl aes-256-cbc -a -iter 1024 -in pic_original.bmp -out encrypt.enc : This command is used for encryption of original image.

time openssl aes-256-cbc -a -d -iter 1024 -in encrypt.enc -out decrypt.decrypt : This command is used for decrypting the encrypted image file.

Encryption and decryption of audio

Commands:

time openssl aes-256-cbc -a -iter 1024 -in audiofile.mp3 -out encrypt.enc : This command is used for encryption of audio information

time openssl aes-256-cbc -a -d -iter 1024 -in encrypt.enc -out decrypt.decrypt : This command is used for decryption of audio information.

III. Result and Discussion

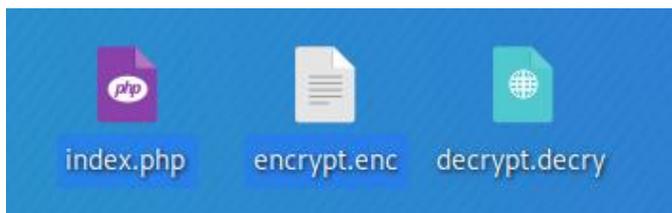
Text File

The original text file is encrypted and decrypted using the command and also time for execution is also calculated.

```
root@kali:~/Desktop# time openssl aes-256-cbc -a -iter 1024 -in index.php -out encryp.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:

real    0m4.927s
user    0m0.011s
sys     0m0.000s
root@kali:~/Desktop# time openssl aes-256-cbc -a -d -iter 1024 -in encryp.enc -out decrypt.decrypt
enter aes-256-cbc decryption password:

real    0m2.435s
user    0m0.011s
sys     0m0.000s
root@kali:~/Desktop#
```



Original is taken which is then converted into encrypted file using AES algorithm. We can see the encrypted file which is not readable and after that the encrypted file is decrypted using the same AES algorithm and by giving the correct key or password we get back the original file

Original file

```
File Edit Search View Document Help
Warning, you are using the root account, you may harm your system.

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
width: 100%;
padding: 12px 20px;
margin: 8px 0;
display: inline-block;
border: 1px solid #ccc;
box-sizing: border-box;

/* Set a style for all buttons */
button {
background-color: #4CAF50;
color: white;
padding: 14px 20px;
margin: 8px 0;

```

Encrypted file

```
File Edit Search View Document Help
Warning, you are using the root account, you may harm your system.

U2FsdGkx18LpN917+wLSC1mgTpryV1AMHDKw2jIfTvmh1j6R190qjT64K7/Qta
DEwFHHGRm/OVlytDkhVdxLMhLFfclVEbSeX4L/Mh4Kpyslx6NSb0ipfAzufLsy
AH0lV4o+xrGey4+xJbcWoz/OKgCLjvSxs8Q1I9fqs5iQku3N2kZNBC6y1EL0DWf
GgUmIIby0aw8PgTBFmZlq0EZpodwh/mpt8Uf0g2Pduz9Hltdm6iEycxrv85Pcf9D
2kONxwLuS+/eiVi0YyEGE0zpe5Jd2XmZrF3A18cPSPMhrGY30BL8A0X0w7SaLDt2
0HFJ0aGIDonCP+nRm6VmJkANZP4Tpmx9X/zw9uLFPMMVOPWpVKeoD67+5f4MX1Q
M4IVamtX77rDwLz9t4p2Yor0FL0cuz6/f8bPaW9Lvs9yEg3ny0Xc6Vy7geXUYK
ILqW20V0FkvS2rvheofq22AHdg2grmXRjXx1QJn4rnFMW/LP66N0vcLm1GjT25GX
xeE14AFvcDVLcZwFceIwXULXJfQ5Z4MrP1dLwHdd8F8w9WvRACbg5Z4Kn1fiJPs
8Lyh6idaqTQRDnSgxfJepK+hKooG25nVgreTVRuB36heN0r8EQ2eWfVwClwG7jad
os0osx7+A6PvIW5ly615rx0CwW7EZ5p33Tj1pgSkkSPFaiqaIQ457rVu5t44+ZEt
kVoPP5WbgAf/NHSPhdme/uKq/ApxsZyxJonPV2swm2cySUGkOTFaC8NDJviHHPH
/HYVfTeoCkMbv8gTUuGOS08daA5pDjwY8l3Muatl1DeItnjvLuZD32tj0ehjCa
itnF11GVstH+KX0p62j0612R1QmTvboLyeYnXh84pDm02kf5DxcwiT2HBxG1D
LqVUAIJXcJbHe0Uvdx75dHwdB9kjNPSPTAvHNKSvDU2BmVADjvZALoKUXYOAE
Ld+tZztFrKXUxgBoaBsD7bJ8NSnQUnnXOST4xAMl2HiC+8cWBSustEumVp81o
Sn01xGfzE38j98SR0h2VgPtiaDwJrnjfcA3yteUP8F04Mx2buEHMjdn0DuYz
FFMsgeefK16MK9EnFLOR4EUC3btC5fwCiwI4it6m6sS/MTS3zCLCjrp0y6S3JNT
O+NM1xZXInupB673TxxFyzR9P5Ej5jrzFzkgQu5qqNdXuiPuMod8QPKORnrvn
mtC3umJn0YzgalBcn37IrTgydzmBgfTPP3B11+AubgXHKjglPIQsHg++0GbiZ8
kYwdIKlJk0GTZ/gWlJrtGmK12i3A9Lz+90s1YHSLQv6j3a354kZwEBE/K3+YpvX50
DAdE8NHu9PtPFQlRB64BSWjdcAQJJuY2jMKD18tYrcuwwAurL09W+WBX1Bv1
1chP1LYIIsP1KB2pNyBIPOEjRSMFCHGTcd0gmQxnjkt0DAFka7fhyGnz1g05BI
```

Decrypted file

```
Warning, you are using the root account, you may harm your system.
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
width: 100%;
padding: 12px 20px;
margin: 8px 0;
display: inline-block;
border: 1px solid #ccc;
box-sizing: border-box;
}

/* Set a style for all buttons */
button {
background-color: #4CAF50;
color: white;
padding: 14px 20px;
margin: 8px 0;
```

Image

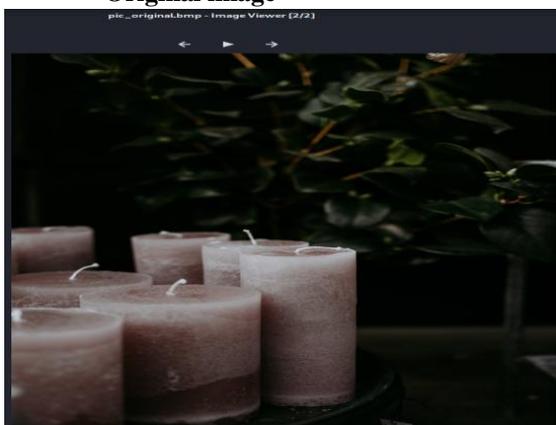
The original image is encrypted to non readable format and then converted back to its original form by decrypting it using AES algorithm and also time of execution is also calculated.

```
root@kali:~# cd Desktop
root@kali:~/Desktop# time openssl aes-256-cbc -a -iter 1024 -in pic_origina
l.jpg -out encrypt.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:

real    0m5.227s
user    0m0.009s
sys     0m0.014s
root@kali:~/Desktop# time openssl aes-256-cbc -a -d -iter 1024 -in encrypt.
enc -out decrypt.decry
enter aes-256-cbc decryption password:

real    0m3.519s
user    0m0.016s
sys     0m0.004s
root@kali:~/Desktop#
```

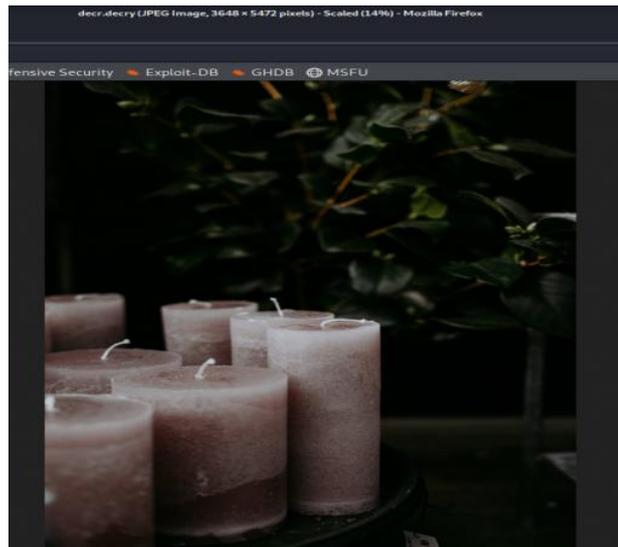
Original image



Encrypted image

```
Warning, you are using the root account, you may harm your system.
J2FsdGvKX1+Ft/kQUTWXHNawz97ca4s7SoCMxIH5*3YiqJhgmx2LGFdk0*EB4pjp
a2BkJUNCKh3ptNQK69nBmM60xgnpLah7oVujEYbK/3E0ExrLRV75SanqeUWqVu68z
TMCq7e05+Azh7PCDwo621CdzJDDJ++kxkA1nrWYOGCxn72wpgg5xQkdd4gDY+vDA
75B/c/brXnA3Xwjf1853oafrnqmqj8LVhSpGrXx860+dWfB2Tb3Q4E0L38ioV8UC4
AIfv3UGcNweqgov2VR06RL84mWC9DQJ35Woa50ppd54P5I0vSvN5KpFXs4LE8BUS
HKi/9bWSzy/AzUzIVmAkIP1Zq7vVMysIXMK/OvWa08UU7qKyxBKAqTXXH9xsCMCm
UNJE3AoBHB+pWYqST/dv6nyx6J0fuctYkWwvSUno2rmtJImQ4aoHx+Y8g51FF
b5Bm25oFLfDEEJaQMeazNA+Tg6pJzL2paLf1CV7Jic9sf3GwRoQ11Vr5TckVzTfs
xLfn4H0V4BbHhPK4i1FEYwZXRkRkDU1b1JstgiaK12PvOxVIOuo6i1754aEK18t
FjJtwDpgxkpIs31PzskWwL7YptwXhwA1CUXxd7rX/t7jI1rwwPqYq/+QLHZfha
/zLMPty6TbtKcWJV1WkS1kIB8205fa/6vQy94GyNIud6toXAWBP8V3yvQcnXGk
bUIaiznfsRtLRamwntZnci7/tacESpasWJQqgdkZftLmbX410iORzH+rcG/UHO
bHKbWwNw+pceAaeY34xtXZJp7xPw2qClddX5cboYGESHCqwT4FFdTOmtD9Y+It1N
qsmW05e11zkB3S+G6tHeAveppj9mieDQTJJsK8H9P8y/4ncDGf4kuFhurFvMnj
WUtPr1JxPAC5iHF72WC9hfF9NpFIx4dqmM5Q4mPHdYb+o60GK36f8Shfz0/SS+c
90S63dvT7ReuhFyTT2jmR8n3jmeWcdouNwA4z0M1EccyE6jiNxdzSI/mT2W7aNI
glbbu8opHbC132yNMBMoT4jTUhdzGqSSondfmb8A750EduR24VGXT+wg9vuu/gANJ
Pb1w3I/Q/PyCwCcUwJqLp/r3csIHeYkRz90mLBGvtz7jSBcF0y2pDvqYYA/hA
SN8jktLDbQLdm0g10AUP8VHSD50a2gkjiw0LwiZuqiFjkFH36/F3LXMH5EvuSyw
MmmtQ5Ipc0Lm4DPC3zUoAv0y90nmdaFUBQVajgrJ07nimudFFjwdeoo1dY9LBDse
b7IOUla88Y78YTz0k4RWB/TRE/utFW4I/2yY8wiXmG3xL3gXwKV+oiBkPQF2y7j
tN0RL5h+z0H0nrC99Qe8ankxSLQLPawee85EKIN9/K9GvHsy/eg+8w28kUx467Xe
```

Decrypted image



AUDIO

The original audio file is encrypted and decrypted using the command and also time of execution is calculated.

```
root@kali:~/Desktop# time openssl aes-256-cbc -a -iter 1024 -in audiofile.mp3 -out encry.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:

real    0m4.196s
user    0m0.004s
sys     0m0.008s
root@kali:~/Desktop# time openssl aes-256-cbc -a -d -iter 1024 -in encry.enc -out decr.decry
enter aes-256-cbc decryption password:

real    0m2.591s
user    0m0.010s
sys     0m0.000s
root@kali:~/Desktop#
```



Time taken for Encryption of text file is maximum followed by image and audio file. Decryption of files took more time compared to encryption.

IV. Conclusion

In this it shows s the successful encryption and decryption of normal text, file and image using AES algorithm in openssl tool. It gives better security against unauthorized access It also give the time consumption for the encryption and decryption to occur. The algorithm guarantees secure end to end transfer of data without any corrupt data. In future the work may be extended to audio and video encryption using openssl tool or different tool and comparing the time required for encryption and decryption.

References

- [1]. International journal of Scientific and Engineering Research, Volume 7, Issue 2, February-2016 ISSN 2229-5518
- [2]. File Encryption, Decryption Using AES Algorithm in Android Phone
- [3]. A novel sensitive image encryption algorithm based on the Zaslavsky chaotic map
- [4]. International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 6 Issue 4 April 2007 , Page No. 20915-20919
- [5]. W. Stallings, Cryptography and Network Security, 4th Ed, pp. 58-309, Prentice Hall, 2005.
- [6]. W. Millan, "How to Improve the Nonlinearity of Bijective S-boxes," Lecture Notes in Computer Science, Vol. 1438, pp.181 - 192, Berlin: Springer-Verlag, 1998.

- [7]. W. Stallings, *Cryptography and Network Security*, 4th Ed, pp. 58-309, Prentice Hall, 2005.
- [8]. AtulKahate, “computer –based symmetric key cryptographic algorithm ” , in *cryptography and Network Security* , 3th Ed. New Delhi McGraw-Hill , pp.130-141.
- [9]. Sourabh Singh, Anurajjain ,(2013, May).”An Enhanced Text To Image Encryption Technique using RGB substitution and AES” , *International journal of Engineering Trends and Technology (IJETT)* volume -4,issue-5,pp.2108-2112.
- [10]. [https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard#:~:text=The%20Advanced%20Encryption%20Standard%20\(AES,cybersecurity%20and%20electronic%20data%20protection.](https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard#:~:text=The%20Advanced%20Encryption%20Standard%20(AES,cybersecurity%20and%20electronic%20data%20protection.)
- [11]. https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.html
- [12]. <https://www.openssl.org/>
- [13]. <https://www.gitmemory.com/issue/elasticdog/transcrypt/59/491230983>
- [14]. <http://myhackingjournal.blogspot.com/2016/10/encryption-and-decryption-of-images.html>

Nikhil Anand. “Using Aes Algorithm Encryption and Decryption of Text File, Image and Audio in Openssl and Time Calculation for Execution.” *IOSR Journal of Computer Engineering (IOSR-JCE)*, 22(6), 2020, pp. 39-44.