G303

The Implementation of the Advanced Encryption Standard (AES) Encryption Algorithm For Computer File Security

Mokhammad Hendayun, Iwan Abadi, Marulan Heryanto Samosir
Departement of Informatics Engineering
Langlangbuana University (UNLA)
Bandung, Indonesia
hendayun@aol.de , Iwan.abadi69@gmail.com

Abstract-A file is related collection of record or file, that can be translated into archive/data and then stored in secondary storage, such as magnetic disk and magnetic tape. File may contain texts, images, video, audio or combinations of them. Security of computer files is essential to implement, because the files can be stolen by unauthorized person. Therefore we need a method to secure files, method of securing files that are often used include cryptographic. By using the methods of cryptography, files will be encrypted by means of randomizing. Encryption will change the record or text (plain text) into record that can not be read (cipher text). There are several algorithms that can be used to encrypt the file, one of them is Advanced Encryption standad (AES). Hence in this study will be discussed the solutions to problems with file security algorithm using AES, so that the file is safe from people who are not responsible.

Keywords: File Security, Cryptographic, Encription, plain text, cipher text

I. INTRODUCTION

The computer file security is a very important point in the use file on information technology. Processing technology of the computer file can be done from anywhere, because, the communication can be done with the online system. Therefore, how to keeping a file of attack or interference from parties who are not authorized, such things should be notice us. Therefore necessary to secure a methods of the computer files, on method of the computer file security is cryptograph. Cryptography is the science that studies how to keep secure data or messages, from sender to reciver without experiencing interference of the intruder. The security aspects of file security is confidentiality, integrity, availability and authentication. Main process in cryptograph is encryption and decryption. Many of the algorithm with used in cryptography, one of the algorithm is Advanced Encryption Standard (AES).

II. BASIC THEORY

1. Encryption and Decryption

Main process in cryptograph is encryption and decryption. The Encryption is changing the original message (plaintext) into a coded message (ciphertext), while the decryption is to restore the ciphertext into the original message (plaintext) [1]. Overview of the cryptography process can be seen in figure 1 below:

Figure 1 Cryptography Process

2. Advanced Encryption Standard (AES)

The AES is a symetric blok cipher that uses 128 bit, 192 bit or 256 bit keys [2]. For AES 128 bit, The key length and block size can be chosen independently. Each block is encrypted in a certain number of rounds.

3. File

When all the records representing entities of one type are collected together, we call the aggregation a file. Files can be viewed as both logical and physical entities [3]. The file can also be translated archive or data with stored in the storage computer. A file is related collection of record or file, that can be translated into archive/data and then stored in secondary storage, such as magnetic disk and magnetic tape. File may contain texts, images, video, audio or combinations of them. In the information technology necessary to the security file of interference person who do not have access.

III. ANALYSIS AND DESIGN

1. System Description
   The system build called Maru File Protector (MFP). The mechanism of MFP system is:

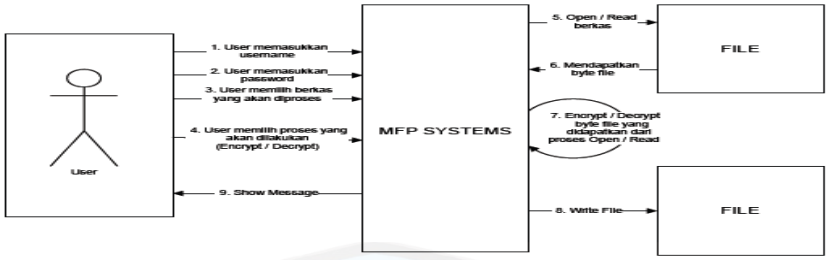

Figure 2 Description of System

TABLE 1 MECHANISM OF SYSTEM

NO	MECHANISM
1	Input username by user
2	Input password by user
3	User memilih file yang akan diproses
4	User memilih proses yang akan dilakukan (enkripsi atau dekripsi)
5	File yang telah dipilih akan dibuka kemudian dibaca untuk mendapatkan byte file
6	Get of byte file
7	Melakukan enkripsi/dekripsi byte file
8	Setelah prose file dienkripsi/didekripsi maka hasilnya dituliskan ke dalam file
9	Show message

2. User of System
   User of system are those that interact directly with the software. In this developed system that interact with the software are user, in the system is referred to as actors.
3. Functional Requirements
   Requirements analysis results in the specification of software's operational characteristics; indicates software's interface with other system elements; and establishes constraints that software must meet [4]. Functional requirements is description of what the system should do. Functional requirements of the system can be seen from the table below:

TABLE 2 FUNCTIONAL REQUIRMENTS

REQUIREMENT NUMBER	REQUIREMENT NAME	DESCRIPTION
MFP-10	Encryption	The system can make changes to the files in the ciphertext
MFP-20	Decryption	The system can make changes to the ciphertext files in the plaintext

Of the functional requirements, the next step is to translate requirements into use cases. Use cases are scenario for understanding system requirements. A use case model can be instrumental in project development, planning, and documentation of systems requirements [5].

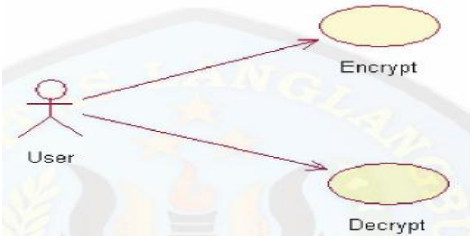

Figure 3 Use cases Diagram

4. Advanced Encryption Standard (AES) Algorithm
   The AES class consists of methods and variables, the following more detailed:

a. AES()
   The method is constructor of AES class
   Input : byte[]arr_keys
   Output :
   Proses : - setting the size of the variable block_state dan block_key
   - Call and running of the method build_sBox(), build__InvsBox(), and build_rCon().

b. setKeys()
   This Method aims to setting of the keys is used:
   Input : byte[]arr_keys
   Output :
   Proses : - Setting variable block_key from arr_keys
   - Form upa-keys or expended keys.

c. Encrypt()
   This Method aims to encryption in the file given to length of variation inputs.
   Input : byte[]arr_plaintext
   Output : byte[]
   Proses : - Get the length from arr_palintext
   - Create an array of result with the length of arr_block*16
   - Perform looping round total block array (arr_block) to encryption for the arr_plaintext

d. Decrypt()

This Method aims to decryption in the file given to length of variation inputs.
Input : byte[]arr_cipher
Output : byte[]
Proses : - Get the length from arr_cipher
- Create an array of result with the length of arr_block*16
- Perform looping round total block array (arr_block) to dekription for the arr_cipher.

e. EncryptByte()
This Method aims to encryption file with length 16 byte
Input : byte[]arr_state
Output : byte[]
Proses : - Conversion from form arrays to form block, setting its value to perform variable block_state
- Call method AddRoundKey() with input value = 0
- Perform looping round Nr – 1, and call method SubBytes(), ShiftRows(), MixColumns(), and AddRoundkey(). For method AddRoundKey() input value is iNr + 1.
- Call method SubBytes()
- Call method ShiftRows()
- Call method AddRoundkey() with input value is Nr.

f. DecryptByte()
This Method aims to decryption file with length 16 byte
Input : byte[]arr_state
Output : byte[]
Proses : - Conversion from form arrays to form block, setting its value to perform variable block_state
: - Call method Addroundkey() with input value is Nr
- Call method InvShiftRows()Call method InvsubBytes()
- Perform looping round iNr > 1, and call method AddRoundKey(), InvMixColumns(), InvShiftRows(), and InvSubBytes(). For method AddRoundKey() input value is iNr-1
- Call method AddRoundkey() with input value is 0.

g. build_sBox()
This Method aims to forming a series of constant matrix called sBox
Input :
Output :
Proses : Setting of value from the variable sBox with matrix 16 X 16

h. build_InvsBox()
This Method aims to forming a series of the substitution box inverse constant matrix called Inv s_Box.
Input :
Output :
Proses : Setting of value from variable iv_sBox and with matrix 16 X16

i. build_rCon()
This Method aims to forming a series of constant matrix called rCon
Input :
Output :
Proses : Setting value from variable rCon and with matrix (Nr + 1) X 4

j. SubBytes()
This Method aims to perform substitution between state box with sBox substitution.
Input :
Output :
Proses : Perform loop for iSubBytes < 16 for sBox substitution block_state

k. InvSubBytes()
This Method aims to perform inverse for substitution between state box with sBox substitution box.
Input :
Output :
Proses : Perform loop for iSubBytes < 16 for substitution block_state to iv_sBox.

l. ShiftRows()
This Method aims to perform cyclic rotation on block_state
Input :
Output :
Proses : Perform loop for iShiftrows < 16 for cyclic rotation.

m. InvShiftRows()
n. This Method aims to perform cyclic rotation on block_state
Input :
Output :
Proses : Perform loop for iShiftRows < 16 for cyclic rotation.

o. MixColums()
This Method aims to do the mixing between block_state with Transformation sBox.
Input :
Output :
Proses : Perform loop for iMixColumns < 4 for do mixing between block_state with Transformation sBox.

p. InvMixColumns()
This Method aims to do the mixing between block_state block_state with Transformation sBox inverse.

Input :
Output :
Proses : Perform loop for iMixColumn < 4 for do mixing between block_state with Transformation sBox inverse.

q. AddRoundKey()
This Method aims to perform XOR between block_state with upa-key in the order. Input : int ipos
Output :
Proses : Perform loop for iAddRoundKey < 16 for merge between block_state with upa-key in the order position the iPos.

r. build_Expansionkeys()
This Method aims to establish a series up-key that will be used for the method AddRoundKey()
Input :
Output :
Proses : - Setting keys for user input is the block_key ke block_w with position 0
- Perform loop iExpand < Nr, for get the upa-key.

s. GFMult()
This Method aims to perform polynomial multiplication.
Input : - byte b
- byte p
Output : int
Proses : - If the multiplier (p) with value 0X01, then return b value.
- If the multiplier (p) with value 0X02, then check b, if value < 128 then give the value of the value of bis shifted 1 bit to the left. If it is greater than 128, then the value of the results in shifted 1 bit to the left at XOR with the multiplier p.
- If the multiplier (p) with value 0X03, then a the result value back to GFMult(b, 0X02) in XOR with GFMult(b, 0X01)
- If multiplier (p) with value 0X09, then a the result value back to GFMult(GFMult(GFMult(b, 0X02), 0X02), 0X02) in XOR with GFMult (b, 0X01)
- If the multiplier (p) with value 0X0b, then a the result value back to GFMult(GFMult(GFMult(b, 0X02), 0X02), 0X02) in XOR with GFMult(b, 0X03)
- If the multiplier (p) with value 0X0d, then a the result value back to GFMult(GFMult(GFMult(b, 0X02), 0X02), 0X02) XOR GFMult(GFMult(b, 0X02), 0X02) and in XOR with GFMult(b, 0X01)
- If the multiplier (p) with value 0X0e, then a the result value back to GFMult(GFMult(GFMult(b, 0X02), 0X02), 0X02) XOR GFMult(GFMult(b, 0X02), 0X02) and in XOR with GFMult(b, 0X02)

t. PacketdTo16()
This Method aims to do package to an array of length 16
Input : byte[]arr
Output : byte[]
Proses : Perform loop for iPacked < iSize to move the conents of arr to variable tmpPackedTo16.

u. SetArrByPos()
This Method aims to do setting 16 byte from array arr_pos
Input : - byte[]arr_input
- int arr_pos
- ref byte[]arr_return
Output :
Proses : Perform loop for iArr smaller than the size of the array arr_input to perform setting of the arr_return in the position of the arr_pos.

v. GetArrByPos()
This Method aims to get 16 byte from the array in order with arr_pos
Input : - byte[]arr_input
- Int arr_pos
Output : byte[]
Proses : Perform loop for iArr value smaller than the size iBatas to get the array of the arr_input in order with arr_pos

5. The File Structure That Have Been Encrypted
Encrypted file structure is as follows:

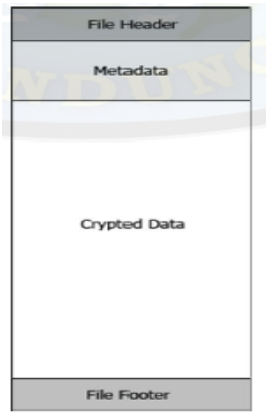

Figure 4 File Structure

IV. IMPLENTATION

The mai menu of the software is to display the encryption and decryption.

Figure 5 The Main Menu

1. The Encryption Process
   The encryption process steps are:
   a. Fill in your username
   b. Fill in your password
   c. Press the button encrypt
   d. Choose a file to the encrypt

2. The Decryption Process
   The decryption process steps are:
   a. Fill in your username
   b. Fill in your password
   c. Press the button decrypt
   d. Chose a file to the decrypt

V. CONCLUSION

The conclusion is:

1. The file security is assured if the encryption is done before the file in sent in a computer network.
2. AES algorithm, is a cryptographic algorithm that produces a good files security.
3. By performing encryption, file security aspects will be fulfiled.

REFERENCES

[1] Munir Rinaldi,"Cryptography", INFORMATIKA, 2006

[2] Brenton Chris, Hunt Cameron,"Network Security", SYBEX Inc. 2003

[3] Harbron R. Thomas, "File Systems; Structures and Algorithms", Prantice-Hall, Inc., 1988

[4] Pressman S Rogers, "Software Engineering", McGraw-Hill International Edition, 2005

[5] Bahrami Ali, "Object Oriented System Development; using the unified modeling language", The Mcgraw-Hill Book Co, 1999