International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 3.45 (SJIF-2015), e-ISSN: 2455-2585
Volume 4, Issue 5, May-2018

A Proposed System Concept on Enhancing the Encryption and Decryption Method Using USB Mass Storage Device

Mr. Anand Prakash Rawal[1], Mr. Vishal Kumar[2], Mr. Abhishek Kumar Sinha[3], Mr. Madhurendra Purbay[4]

[1] *Department of Information Technology, Institute of Technology, Guru Ghasidas Vishwavidyalaya (A Central University),Bilaspur , rawal96@gmail.com,*

[2] *Department of Information Technology, Institute of Technology, Guru Ghasidas Vishwavidyalaya (A Central University),Bilaspur , vishalkumar08021998@gmail.com*

[3] *Department of Information Technology, Institute of Technology,Guru Ghasidas Vishwavidyalaya (A Central University),Bilaspur , aksinha050@gmail.com*

[4] *Department of Information Technology, Institute of Technology,Guru Ghasidas Vishwavidyalaya (A Central University),Bilaspur , madhurendra7796@gmail.com*

Abstract

We have proposed system that involves the security of data by using the external Universal Storage device (USB) and Personal Identification Number (PIN) using as a key .The model should reduce time complexity by altering the traditional method of AES algorithm. We modified the AES algorithm key generation part with SHA to increase the security part. It reduces the time complexity of traditional AES algorithm. It gives the user more control on selection on his/her password, which were random generated earlier We have found that after modifying the AES algorithm, time complexity of overall encryption/decryption is reduced than traditional method, Key generation of AES algorithm takes more time than the message digest of SHA256, so replacing that part increases the time of encryption/decryption of same file. We have also found that using SHA, intruder can't generate the input (USB' serial number and user's PIN) from output. Authors have found that this model can be used for the securing the file from unauthorized access , can be used for personal data encryption/decryption. Authors have improved the AES algorithm after replacing the key generation part with SHA.

Keywords: AES algorithm, Hash algorithm, USB mass storage device, Cryptography, Symmetric key, 256 bit.

Introduction

From the evolution of mankind, cryptography has evolved with the same tremendous speed. In the earliest time, humankind has been attempting to hide some information to keep to their own or substituting with symbols, numbers and pictures. For some different reasons humans have been always interested in protecting their messages or information from others. Now in the digital information age, we still want to protect or hide some information from others. In recent years of development of computer and information industry has produced various new tools and proposed the models for hiding or protecting the document. Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication and data origin authentication. In data and telecommunications, cryptography is necessary when communicating over any unreliable medium, which includes any network particularly the internet [1]. In this era, we mostly keep and generate the important document in electronic format and protecting them from any mischief is one of the important concerns of this time. So, there must have been to plan in advance to prevent the problems related to the disclosure of information. With the evolution of technology also security measures have been improved with the time. The Enigma Machine, which was employed by the Germans to encrypt the data of warfare and was successfully decrypted by Alan Turing, can be regarded as a striking example of creating and using secured information [2]. With time data transferring method also changed. Before USB came into existence, computers used serial and parallel ports to plug devices into computers and transfer data. Individual ports were used for peripherals such as keyboards, mouse, joysticks and printers. Expansion cards and custom drivers were often required to connect the devices. Parallel ports transferred data at approximately 100 kilobytes per second, whereas serial ports ranged from 115 to more than 450 kilobits per second. Some ports could not run simultaneously [3].

During this span of time, Universal Serial Bus (USB) device is easily available to each and every person. In this proposed model, authors are using Universal Serial Bus (USB) device as a key with the combination of PIN to secure the personal data. There have been various models to secure the personal data but most of them have been used single key to protect the data. In this model user needs USB device and PIN for further operations.

AES Algorithm

The **Advanced Encryption Standard** (**AES**), also known by its original name **Rijndael** is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001 [5]. AES algorithm is faster than Data Encryption Standard (DES) algorithm. In AES algorithm rounds depend on the key. AES uses 10 rounds for 128 bit key length, 12 rounds for 192 bit key length, 14 rounds for 256 bit key length. As in figure 1

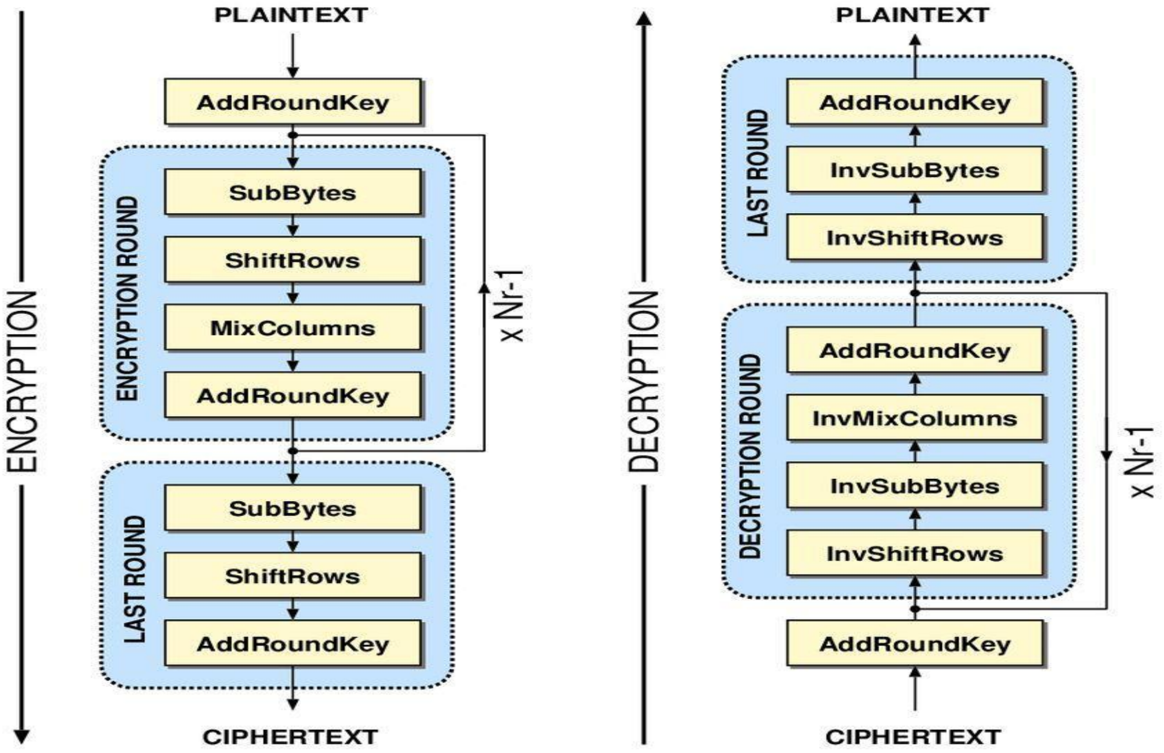


Figure 1.

Overall Structure of AES algorithm:

AES algorithm starts with Add **Round Key** stages followed by N-1th rounds of four stages and a Nth round of three stages. It works in both encryption and decryption with exception that each round of decryption is the inverse of the encryption algorithm. The first N-1th round of the encryption algorithm are:

1. Substitutes bytes
2. Shift Rows
3. Mix Columns
4. Add Round Key

The Nth round just does not add the **Mix Columns** stage.

The first N-1th rounds of the decryption algorithm are:

1. Inverse Shift Rows
2. Inverse Substitutes bytes
3. Inverse Add Round Key
4. Inverse Mix Columns

Again, in Nth round it does not add the Inverse

SHA 256 Algorithms

SHA stands for Secure Hashing Algorithm. It has four variant SHA-0, SHA-1, SHA-2 and SHA-3. SHA-256 is the family member of SHA-2. SHA-256 takes 512-bit blocks, whereas SHA-256 expresses the 512-bit blocks in sixteen different part of length of 32-bit words (32-bit *16 = 512 bit). All five of the algorithms are iterative, one-way hash functions that can process a message to produce a condensed representation called a message digest [7]. Before hash computation begins; message shall be padded to make multiple of 512-bits.SHA convert arbitrary length to a fixed length. It is almost impossible to find the input value using the hash value. It is collision resistant; this property means it should be hard to find two different inputs of any length that result in the same hash. This property is also referred to as collision free hash function [8]

Methodology

This Methodology is based upon the problem which was earlier presented in the AES algorithm and we overcome that problem in this project. AES algorithm uses the symmetric key. In this project, we are indulging an external USB mass storage device which gives another layer of security to the AES algorithm. We are taking the USB's serial number, which is unique to each and every USB device and taking the PIN from the user and concatenating them in the predefined manner and passing them through the SHA-256, which has very different and special properties like collision resistant, and it produces the fixed length of output which is very suitable for our project. Then after getting the output from the SHA-256, we are passing them to the AES rounds to generate the encrypted/decrypted file as per users' need [20].

This procedure is mainly divided into three modules.

1. Login
   a. First time credential input
   b. Selection of Username and Password
   c. Login to the system

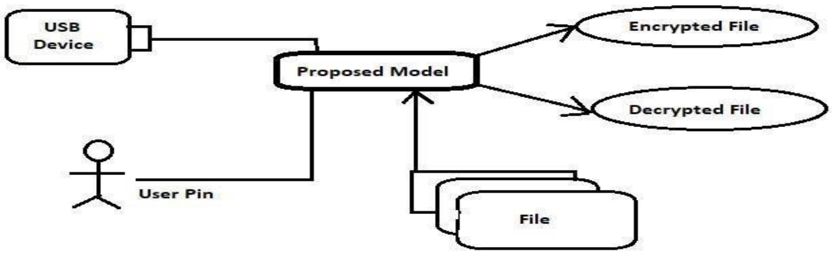
2. Encryption
   a. Selection of file
   b. Insertion of an external device
   c. Insertion of user's PIN
   d. Encryption of file

3. Decryption
   a. Selection of encrypted file
   b. Insertion of an external device
   c. Insertion of user's PIN
   d. Decryption of file

Working Procedure of Algorithm

In this model, AES algorithm works as same procedure as it works earlier. The only difference is of key expansion and key generation. As AES algorithm generates its own key but in this model, the key is generated by SHA-256 algorithm. Here User's pin and USB's serial number is concatenated no matter of what length it is in, by taking first user's pin and then USB's serial number and then it is passed through SHA-256 algorithm. SHA-256 algorithm breaks it into 512 bits size of data. If the size of the concatenated text is less than 512 bits then it is padded with zero bits. This 512 bits size of text is converted into the fixed length encrypted text of 256 bits by SHA -256 algorithm which is used as the cipher key and send it to AES algorithm. AES algorithm uses this 256 bits cipher key in its every round for encryption of file instead of using its own generated key. Generally, AES-256 algorithm takes 14 round to generate cipher key which is to be used for encryption of file. But by using SHA-256 algorithm for key generation save the time and reduced the time complexity of the model. As in figure 2

Figure 2.



Current system and application

For maintaining the authentic access and use of data, we've taken the user's credentials for the first time of running the system. First module has main works to stop the unauthorized access to next module. In this module, the data which are inserted are stored in a text in the file in encrypted form. The details that the user inserts are passed through SHA-256 algorithm which encrypts the user's details and stores in the text file. This text is created if the user has created his ID for the first time otherwise it will not allow creating his ID. However, the user will get chance to reset his password by clicking on forgot the password.

The second module is the Home page which contains two buttons named as 'Encryption button' and 'Decryption button'. On clicking either of two buttons, a user can go to Encryption page and Decryption page respectively. This page consists of encryption module and decryption module.

In Third module, a user has to enter the PIN and an external USB storage device. USB device is used for their unique serial number. After taking both inputs, we are passing it through the SHA-256 algorithm to generate a fixed length password. This fixed length password is irreversible in nature which cannot be reversed back to find out the user pin and the serial number of pen drive that user has inserted. Now fixed length password will pass to encrypt/decrypt module.

In Fourth module, we have to take the file from the system and fixed length password which has been generated in the module 3 and it is passed through AES algorithm in order to generate the encrypted file/data. The password is transferred to the security program without any protection, the attacker can find out the password on the communication between the security program and USB flash drive. To solve this problem, the security program needs to use an encryption function or a hash function [4].

In Fifth module, we have to take the encrypted file from the system and fixed length password generated the module 3 is passed through AES Decryption algorithm to generate the original/decrypted file. Decrypted file can be saved into the user drive as per his desire. As in figure 3.

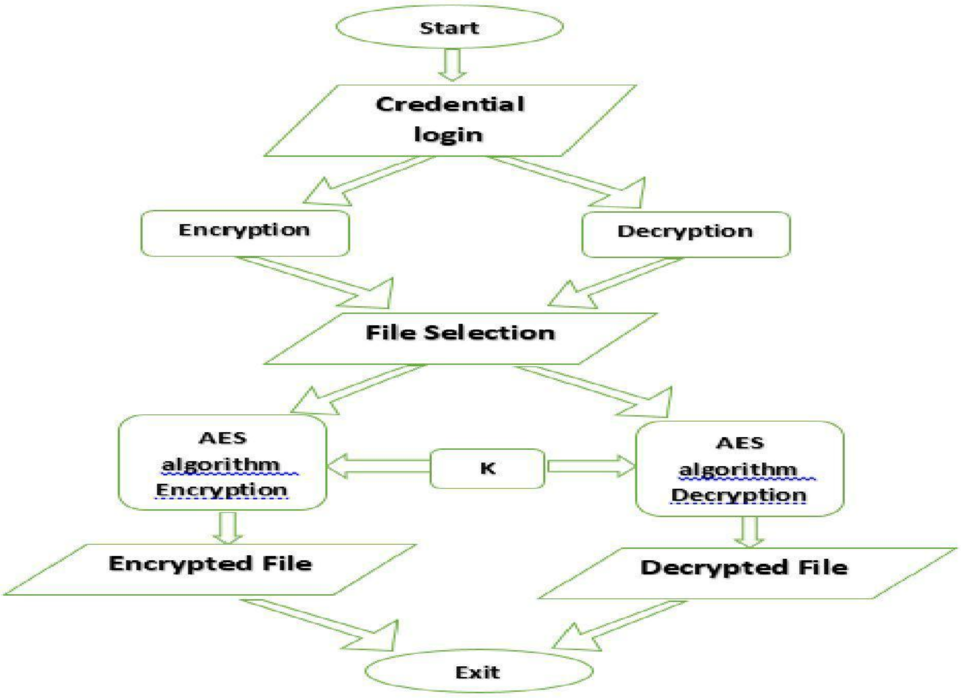


Figure 3. Flow diagram of application system

Result and Conclusion

Using this model increases the security and reduces the time complexity of the model.

The proposed system increases the efficiency of the AES algorithm in term of time complexity by replacing the key generation part of AES algorithm with SHA algorithm. It also provides the user to have their own password without remembering what it was before that was randomly generated by AES algorithm. Use of external USB device provides more security to the user's personal data as without external USB device the file won't be decrypted. Providing the user with interface helps him to handle the software comfortably. User can use this application for the securing their sensitive file and it provides better security compare to other file encryption software.

References

[1] M.Pitchaiah, Philemon Daniel, Praveen "Implementation of Advanced Encryption Standard Algorithm", International Journal of Scientific & Engineering Research Volume 3, Issue 3, March-2012

[2]Sebag–Montefiore, H. (2011). Enigma: The Battle for the Code. Orion. p. 576. ISBN 9781780221236

[3]http://www.allusb.com/usb-history

[4]https://en.wikipedia.org/wiki/Hash_function

[5]https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf

[6]https://crypto.stackexchange.com/questions/8043/aes-addroundkey

[7]https://csrc.nist.gov/csrc/media/publications/fips/180/3/ archive/2008-10-31/ documents/fips180-3_final.pdf

[8]https://en.wikipedia.org/wiki/Login [9]www.ijeert.org/pdf/v2-i4/10.pdf

[10]https://en.wikipedia.org/wiki/Encryption

[11]https://www.techopedia.com/definition/1773/decrypti on

[12]Shtewi, A.A., Hasan, B. E. M., Hegazy, A. El F. A. (2010). An Efficient Modified Advanced Encryption Standard (MAES) Adapted for Image Cryptosystems. International Journal of Computer Science and Network Security (IJCSNS),10(2),226-232.http://paper.ijcsns.org/07 _book/201002/20100234.pdf

[13]https://www.omicsonline.org/open-access/implementa tion-of-advanced-encryption-standard-algorithm-with-key length-of-256-bits-for-preventing-data-loss-in-an-organiza tion-0976-4860-1000183.php?aid=88504

[14]http://www.ijsrp.org/research-paper-1301/ijsrp-p1315. pdf

[15]http://www.jatit.org/volumes/Vol86No2/4Vol86No2. pdf

[16] https://www.ijsr.net/archive/v3i6/MDIwMTQxOA==.pdf

[17] https://pdfs.semanticscholar.org/0177/00ba0459185e3670 d3317d54f137c4fef8b6.pdf

[18] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1. 687.6598&rep=rep1&type=pdf

[19]https://www.tutorialspoint.com/cryptography/cryptograph y_hash_functions.htm

[20]http://www.ijeert.org/pdf/v2-i4/10.pdf