International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 12, Issue 5 (May 2016), PP.70-75

"Customized Algorithm to Enhance the Computer Data Security Using Hardware Key And Randomization Technique"

Nidhi Desai[1], Amee Chauhan[2]
[1]*M.Tech Student Of Computer Department, Gujarat University.*
[2]*M.Tech Student Of Computer Department, Gujarat University.*

Abstract:- During the last decades, information security has become a major issue. Encrypting and decrypting data have recently been widely investigated and developed because there is a demand for a stronger encryption and decryption which is very hard to crack. Cryptography plays major roles to fulfilment these demands. Nowadays, many of researchers have proposed many of encryption and decryption algorithms such as AES, DES, and others. But most of the proposed algorithms encountered some problems such as password cracking, time taken for encryption and decryption. Password cracking is major issue for just password protected security system. Software based security system is only password protected thus if some gain full access to user's password than it may able to interact with whole system. This can't consider as secure if user's privacy is compromised. Hardware devices have the ability to tie a system or an object together with its users to the physical world. Thus by adapting hardware based security the system becomes more efficient and secured as compared to software based security. This paper describes the customized algorithm along with hardware key and randomization technique; it scrambles the whole structure of data like text, image, audio and video to make it unreadable or unintelligible during transmission. Its main goal is to keep the data secure from unauthorized access.Also shows its performance comparison along with AES and DES algorithms and shows how a hardware key enhances the security levels and provides more transparency.

Keywords:- AES, DES, Customized algorithm, Encryption, Decryption.

I. INTRODUCTION

The objective of this algorithm is to protect sensitive data which are stored on local computer from unauthorized theft. Laws, regulations, corporate standards and guidelines from higher authority define "Private and Sensitive" data as unencrypted computerized information that can identify an individual, combined with a social security number, drivers license or financial account access code, i.e. credit card PIN. And "unencrypted" as meaning when either the identifying information or data element is not encrypted or is encrypted with a key that has also been compromised. Existing security system is just password protected so we can't consider those systems as secure because password provides low level security because of password cracking. To provide high-level security to user we have introduce a Hardware key along with public and private keys. Hardware key requires the serial number of hardware device; every hardware device has the unique serial key so that it can be used as a key and gives one more level protection for the file encryption process. To overcome password cracking problem we have increases the password length up to 50 characters long. Usually password length is 8 to 12 characters long, so password cracking complexity is lower, thus by increasing password length we have increases the complexity of password cracking. We also have uses randomization arrangement technique; this scrambles the whole file structure rather than data. Thus to enhance computer data security we have developed customized algorithm along with hardware key, randomization block technique and increasing password length.

II. TYPES OF CRYPTOSYSTEMS

Two main types of cryptosystems are used to encrypt and decrypt data. These are briefly discussed below.

1. Secret Key Cryptography[8]:

Secret key cryptography is also known as symmetric cryptography. Symmetric cryptography uses the same key to encrypt and decrypt data. In symmetric key cryptosystems, the sender will first encrypt the message with his/her private key, the data is then sent to a recipient. The recipient will use an identical version of the key to decrypt the data which has previously been obtained from the sender. The recipient will use the same key to respond to the recipient. Thus, the same key is used for encrypting and decrypting a file in symmetric key cryptosystems. The advantages of secret key cryptography are:

- Very fast relative to public key cryptography.

- Considered secure, provided the key is relatively strong.
- Widely used and very popular

However, the disadvantages of secret key cryptography are:

- Both party (the sender and recipient) must know the secret key before data transmission, this feature makes systems difficult to use, and the key cannot be transmitted openly because it would compromise the system security.
- The administration of secret key based system may become impossible to manage as the number of keys escalates rapidly if every host needs to communicate to every other host.
- Careful distribution of secret keys is crucial because their distribution is security critical which is the biggest difficulty in using secret key cryptography.
- Secret key cryptography cannot be used for non-repudiation. Non repudiation can only be used with a trusted third party.

2. Public Key Cryptography[8]:

This type of cryptography, which is known as asymmetric cryptography, uses different keys to encrypt and decrypt data. Using public key cryptography, a message sender encrypts data using the recipient's public key; the receiver will then decrypt the data using their own private key. The advantages of public key cryptography are:

- One of the main advantages of public key cryptosystems is that if the public key of one recipient is compromised by an interceptor, only that file can be read, deleted or modified; other file will remain secure as their public keys are different from the compromised public key.
- Data encrypted by a sender with a sender's private key can only be decrypted by the recipient's paired public key.
- No form of secret sharing is required, thus reducing key administration to a minimum.
- The number of keys managed by each user is significantly less compared to secret key cryptography.

However, the disadvantages of public key cryptography are:

- Significantly slower than secret key cryptography.
- The cipher text is much larger than the plaintext, relative to secret key cryptography.
- Considered computationally costly.

A number of public key cryptography were selected for the Advanced Encryption Standards and they offer a variety of benefit to improve the security of computer data, those cryptosystems are identified and evaluated below.

III. Current Advanced Cryptosystems

Some of the advanced cryptosystems are identified and evaluated in order to select a suitable algorithm to developed customized algorithm by modifying it.

1. Rijndael[3]:

The Rijndael algorithm was first adopted in 2000 by the US National Institute of Standards and Technology as the Advanced Encryption Standard (AES). Rijndael is an iterative block cipher which supports variable block length and key length. Variable block and key length could be independently specified as 128, 192 or 256 bit and it has a variable number of iterations of 10, 12 and 14 for key lengths of 128, 192 and 256 respectively. A round transformation of three distinct invertible uniform transformations (layers) is used in Rijndael. A very fast implementation on processors with word length 32 or more, could be achieved by using the different steps of round transformation which could be combined in a single set of table lookups. Rijndael offers a number of advantages. Firstly, it can be implemented to run at a faster speed than a normal block cipher on Pentium processors; however, there are tradeoffs between table size/performance. Secondly, it can also be implemented on a smart card with little amounts of codes to be written. Thirdly, it requires little amount of RAM and uses a small number of cycles; however, there are some ROM/performance tradeoffs.

2. Two fish[7]:

Two fish is a block cipher and was proposed as a candidate for the AES. It is based on the Fiestel structure which is used in most ciphers, a part of bits of intermediate state is transposed to another position without changes applied to them. It is a 128-bit block cipher using 128-, 192-, or 256-bit keys. Two fish is designed to be highly secure and highly flexible, it is well-suited for large microprocessors, 8-bit smart card microprocessors, and dedicated hardware. Two fish supports variable length of key up to 256 bits, its cipher consists of 16 rounds with each round using 2 round keys, and four keys are each used during pre and post-processing. The Two fish operation in the encryption round is key dependant nonlinear substitutions. In contrast to other AES block ciphers, two fish satisfies the involution property resulting in identical data paths for both encryption and decryption. However, due to the common encryption and decryption data path, CED architectures of Two fish are different from other block ciphers. Two fish is efficient for the implementation of a variety of platforms including 32bit processors. It is designed to allow for several layers of performance tradeoffs, depending on the relative importance of encryption speed, key setup memory use and other implementation parameters.

3. MARS[4]:

MARS is a shared key block cipher, the block size of which ranges from 128 to over 400 bits and has a variable key. Its encryption consists of 32 rounds, 8 forward mixing rounds, 16 main keys transformation rounds each using two round keys, and 8 backward-mixing rounds. During pre- and post-processing four rounds keys are added to the input data. Its component is designed to allow extensive analysis to guide a number of implementation choices. MARS is suited to the Type-3 Fiestel network, which offers a trade off speed, strength and suitability for analysis, as it has a block size of 132 bits and word size of 32 bits, and each block size consist of four words. MARS can be secured against cipher text attack as plaintext attacks because it is very symmetric; the last half of the rounds are almost mirror images of the first half.

4. RC6[5]:

RC6 is an evolutionary improvement of RC5 that is designed to meet the requirements of the AES. Like RC5, RC6 makes essential use of data-dependent rotations. New features of RC6 include the use of four working registers instead of two, and the inclusion of integer multiplication as an additional primitive operation. The use of multiplication greatly increases the diffusion achieved per round, allowing for greater security, fewer rounds, and increased throughput. RC6 is a parameterised family of encryption ciphers that essentially use the Fiestel structure; 20 rounds were specified for the AES submission. The round function of RC6 uses variable rotations that are regulated by a quadratic function of the data.

5. Serpent[6]:

Serpent is a 128 block cipher with a block size of up to 256-bit keys. It is an SP network and consists of alternative layer of mixing such as s-box and linear transformation. It has an equivalent of a bit sliced description which makes it very efficient. Serpent is a 128 block cipher with a block size of up to 256-bit keys. It is an SP network and consists of alternative layer of mixing such as s-box and linear transformation. It has an equivalent of a bit sliced description which makes it very efficient. Serpent is a 128 block cipher with a block size of up to 256-bit keys. It is an SP network and consists of alternative layer of mixing such as s-box and linear transformation. It has an equivalent of a bit sliced description which makes it very efficient.

Based on the findings, it is decided to select Rijndael algorithm to create Customized algorithm with extra key by modifying it. The strength it has to protect file contents from hacking, for example, it supports variable block length and key length. It can be implemented at a faster speed than a normal block cipher, and a large number of candidates who tested it; they acknowledge that it has no major weaknesses.

IV. Primitives

Existing algorithm's primitives are limited. They cannot be considering, as secured because it just uses password key to encrypt and decrypt file and those algorithms just encrypts the file's data not the whole file structure. We have uses extra key along with randomization technique and master key for recovery purpose.

1. Randomize Block arrangement Technique:

This technique will first convert the file in part of blocks and then it will randomize these blocks and arrange the blocks in random order and write into another file using randomization algorithm. This technique will scrambles the whole file structure into part of blocks then randomized those blocks and randomly arranges those blocks.

2. Hardware Key:

We have added another key, i.e. Hardware key along with private and public keys. Hardware key require serial number of hardware device, every hardware device has the unique key (for example USB Dongle, or you can even go for the RFID device cards, smart cards) so that it can be used as a key and give one more level protection for the file encryption process.

Advantage of using hardware key for the file encryption is that one cannot decrypt file if he/she do not have USB Dongle even if he/she knows the password of the encrypted file.

(As hardware device we have selects USB Dongle because every person having USB Dongle and also it is cheaper in cost.)

3. Master key:

Master key will be used by the system developers only for recovery purpose. In this key the whole encrypted string will be there, so admin having this key can able to recovered data. Recovery is just in case if lost of hardware device which you have used.

V. Customized Algorithm with above Primitives

The customized algorithm is an attempt to present a new approach for encryption and decryption of files of various formats along with hardware key. The system which implements this customized algorithm automatically detects the serial number of USB Dongle and uses this serial number as hardware key along with public-private keys.

- **Encryption Process:**

In term of encryption process, the algorithm consists of combination of public key cryptography with an extra key. Whole encryption process as follows:

1. **S**elect File
2. Find the type of file
3. Find the size of the file
4. Input size of the block
5. Count number of blocks = size of file / size of block
6. Convert number of blocks into upper round
7. Read bytes of each blocks and store
8. Randomize order of blocks from 1 to Number of blocks
9. Write random order blocks into binary files
10. Store random orders into string
11. Input password key
12. Check length of key if key is less than 10 characters go to
13. Check if dongle is available
14. If dongle is available find serial key
15. For concatenation add fix padding to size of file Random orders, password key, dongle key then merge all of them into string (Padding is necessary while decrypting file)
16. Encrypt concate string
17. Write binary encrypted string into encrypted file with random blocks
18. Save the file.

- **Decryption Process:**

The decryption process involves converting the encrypted data back to its original form for the receiver's understanding. Decryption process is as follows:

1. Select Encrypted File
2. Input Password key
3. Check password key (for encryption and decryption password key will be same)
4. Check if dongle is available, finds its serial number (same for both)
5. Find its concatenated string (after encryption done)
6. Decrypt string
7. Remove padding to get order
8. Convert binary number to string and write it into another file called decrypted file
9. Save File

This algorithm supports various file formats as shown in below table:

File Type	**Formats**
Text Files	.txt file/ .docx file/ .doc file/ .xps file/ .pdf file/ .rtf
Excel[Spread sheets]	.xlsm file/ .xlsx file/ .xls
Presentation files	.ppt/ .pptx/ .pptm
Image files	.jpg/ .bmp/ .gif/ .ico/ .png
Audio files	.mp3

Video files	.3gp/.mp4
Archive files	.rar file/ .zip file

Table 1: Supports various file formats

- Comparison of Customized Algorithm with AES and DES:

Key Factor	**AES**[1][2]	**DES**[1][2]	**Customized Algo**
Key length	8 Char	8 Char	Up to 50 Chars
Cipher Type	Symmetric Block	Symmetric Block	Symmetric Randomised Block
Multiple keys	No	No	Yes
Hardware Key	No	No	Yes
Loop block size	128 bits	64 bits	65536 bits
Recovery	No	No	Yes

Table 2: comparison with customized algorithm

VI. Performance Evaluation

In order to test the performance analysis for any encryption algorithm, the speed plays a major role. In this paper, the proposed algorithm compared with AES and DES algorithm in term of the speed in for Encryption process with different perspectives. The speed of the algorithm can be characterized by measuring the time required for encryption and decryption. This parameter is measured for Customized algorithm, AES and DES [1].

- **Speed Analysis with respect to different File Size:**

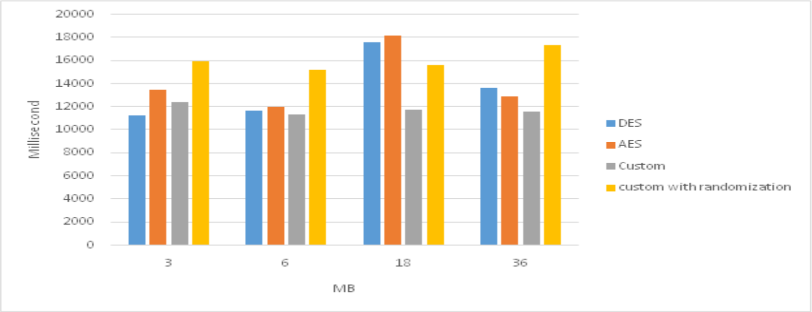

Fig 1. Performance evaluation of customized algo with respect to file size

- **Speed Analysis with respect to different File Types:**

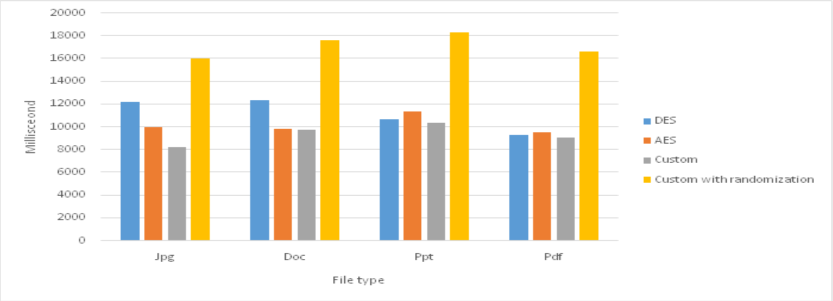

Fig 2. Performance evaluation of customized algo with respect to file type

- **Speed analysis with respect to Randomization Technique:**

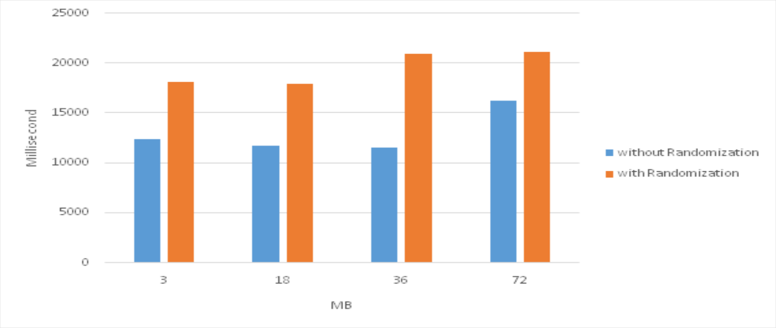

Fig 3. Performance evaluation with respect to Randomization technique.

As well as, the result shows that our customized algorithm takes lesser time to encrypt the file when randomization technique not applied. When Randomization technique applied to customized algorithm it becomes slower.

VII. SECURITY ANAYSIS

In order to test the security level of the proposed algorithm, algorithm should be able to satisfy common security goals. A set of tests and analysis are performed on the algorithm and concluded that it should be able to satisfy common security goals such as:

- **Confidentiality (Data privacy):** The meaning of the message is concealed by encoding it. The message is encrypted using password, dongle serial number and key. And can be decrypted if same password, dongle serial number and key will be provided. Hence data confidentiality is ensured.
- **Data Integrity:** It protects against unintentional alteration of the message. After decryption the content will be same as it was before encryption.

- **Authentication:** The system will identify user by the password and the usb serial number. All the parameters at encryption and decryption time should be same. Also, USB serial number can't be entered manually.

VIII. CONCLUSION

This paper introduced a new approach for encryption and decryption of file. Although there have been many researchers on the cryptography, but most of the existing algorithms have several weaknesses. The Customized algorithm have been tested against different known attacks and proved to be secure against them. Therefore, it can be consider as a good alternative to some applications because of the high level of security and average time needed to encrypt and decrypt a file.

The hardware key used in algorithm contributed towards improving user authentication and overall files security. Performance evaluation has been performed on customized algorithm along with hardware key and randomization technique with AES and DES algorithm and review that customized algorithm is faster than AES and DES. But its performance becomes slower (considered as moderate not too slow) if randomization applied to customized algorithm. Hence customized algorithm is as secured as compared to AES and DES because it not just a password key used it uses hardware key along with private and public which can add another level of security to user.

REFERENCES

[1]. New Comparative Study Between DES, 3DES and AES within Nine Factors, Hamdan.O.Alanazi, B.B.Zaidan, A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani.

[2]. Analysis and Comparison between AES and DES Cryptographic Algorithm, Shraddha Soni, Himani Agrawal, Dr. (Mrs.) Monisha Sharma.

[3]. Rijndael for AES, Joan Daemen, Proton World & Vincent Rijmen, KULeuven,

[4]. The MARS Encryption Algorithm, Carolynn Burwick IBM T. J. Watson Research, Yorktown Heights, NY 10598, USA, Don Coppersmith IBM Corporation, Poughkeepsie, NY 12601, USA, Luke O'Connor IBM Zurich Research, Rueschlikon, Switzerland.

[5]. The RC6 Block Cipher: A simple fast secure AES proposal, Ronald L.Rivest MIT, Matt Robshaw ,Ray Sidney ,Yiqun Lisa Yin RSA Labs.

[6]. Serpent: A Proposal for the Advanced Encryption Standard, Ross Anderson

[7]. Cambridge University, England; Eli Biham Technion, Haifa, Israel; Lars Knudsen University of Bergen, Norway; Twofish: A 128-Bit Block Cipher, Bruce Schneier, John Kelsey,Doug Whiting,David Wagner, Chris Hall.

[8]. How Can Cryptosystem and the Use of Dongles Improve the Security of Backup Files on Servers, Sardar Jaf Northumbria University.