An Improved Symmetric Key Encryption Method Using Randomized Matrix Generation

Karan Padhiyar, Divyang Chauhan, Riya Solanki, and Debabrata swain(✉)

Department of CSE, Pandit Deendayal Energy University, Gandhinagar, India
{Karan.pmtcs21,Divyang.cmtcs21,riya.smtcs21}@sot.pdpu.ac.in

Abstract. Currently, digital data security has appeared as the largest challenge before the society. This concern has become more serious due to the data movement through the unsecured wireless medium. The text format data are mostly targeted by different attackers because of its usage in various finance and other sectors. Different advanced approaches were proposed for securing text data but security concern still remains. In the proposed method a symmetric key cryptographic algorithm is developed for securing the text data. The encryption and decryption key is generated through a set of matrix operations. The Key is generated by the multiplication of random matrices followed by a determinant operation of the same transposed and conversed matrix. The performance of the proposed method is compared with a few existing algorithms using throughput expressed in kilobytes per second. The result analysis has shown that the proposed work with both variations performed well compared to all other discussed algorithms.

Keywords: AES · DES · BLOWFISH · Matrix multiplication

1 Introduction

In the digital world, information security and systems refer to protecting data in terms of availability, confidentiality, and integrity of data. Availability represents features of systems information that can be accessed and utilized on a basis of the specified and desired pattern, as well as when asked in an acceptable manner and according to the system's proper standards. Confidentiality refers to the no-changes, loss of data or authorized users can only disclose the data and information. The guarantee that the information is reliable and correct is known as integrity. It has grown increasingly prone to data exploitation as information systems have gotten more integrated. Here, Cryptographical techniques can be the solution. A cryptographic algorithm is a system that can change data from its original readable form to an encrypted version that prevents fraudsters from accessing the original data. On the other side, Decrypting is the process of turning unrecognizable text back to its original pattern. With the use of certain keys, encryption and decryption are feasible. To protect the data and confidentiality of data we use Cryptography encryption methods. Every encryption technique is designed to make decoding as complicated as possible without the usage of the encryption key.

S. Rajagopal et al. (Eds.): ASCIS 2022, CCIS 1760, pp. 108–117, 2022.
https://doi.org/10.1007/978-3-031-23095-0_8

In the symmetric key cryptography approach, we use the same or single key to perform Encryption and decryption on both sides known as a private key or secret key. Symmetric key cryptographic calculations are isolated into two parts based on the input information: square ciphers and stream ciphers. In square cipher-based frameworks, information is scrambled on a fixed-length bunch of bits called a square, while, in-stream cipher-based frameworks, information is being prepared on a stream of bits [1]. Many methods for encryption with the same key have been developed, each involving various conceptions of cryptographic and temporal complexity other than the concept of processing capabilities during run time. The main purpose of our proposed system is to take a lower total time to execute while retaining the system's complexity. Complex mathematical processes are performed to keep the system's complexity [2].

When compared to asymmetric key methods, the Symmetric Encryption Algorithm is quicker. Because the encryption process is simpler, these algorithms are substantially quicker than asymmetric algorithms in terms of computing. In addition, the Symmetric approach uses less memory than the Asymmetric algorithm [3].

Our Contribution. An effective approach based on symmetric cryptography is established in this suggested system, and its performance is shown based on its required execution time for different sizes of message files.

2 Types of Encryption Technique

Asymmetric cryptographic techniques, symmetric cryptographic techniques, and hash functions are the three most common forms of cryptographic techniques. In the symmetric cryptography technique, we use the same key encryption and decryption of data. Symmetric encryption Some examples include the DES, AES, Blowfish, Carlisle Adams, and Stafford Tavares (CAST) algorithm, SAFER, and IDEA [4].

2.1 AES

AES is a symmetric block cipher technique that encrypts and decrypts using the same key. Joan Daemen and Vincent Rijmen, two Belgian cryptographers [5], invented AES as a variation of the Rijndael block cipher. It is also an iterative cipher since the original input block and key are transformed numerous times before the output is produced. The AES standard specifies that only 128-bit blocks can be accepted, with key sizes of 128, 192, and 256 specified to have the following features: Efficiency in software and hardware, To resist all known attacks, Simplicity in design, and Speed. Ibtihal Mohamed A. Fadul et al. [6] advocated using two secret keys to improve the security of AES. To boost the security strength, the second key is used in encryption and decryption on both sides. Reena Mehla et al. [7] suggested their work to improve the shift row transformation and key expansion of the algorithm to make it more resistant to assaults. Their suggested approach speeds up the picture of encryption and outperforms AES. Faisal Riaz et al. [8] worked together to improve AES by utilizing the DES. They updated the AES algorithm by replacing the Mix Columns phase with the Permutation step. The suggested approach may be used

to encrypt both text and images. Aparna V S et al. [9] proposed the AES technique for secure data transfer. The whole algorithm is coded using MATLAB software. Sumira Hameed et al. [8] also worked on improving AES by utilizing the DES algorithm. Instead of utilizing the Mix Columns step, they updated the AES by employing the Permutation phase. The suggested approach may be used to encrypt both text and graphics. In [10], the author introduced a new approach to AES named revised AES (AES-R). They conclude that AES-R can be the option of the AES because its performance is closed to the traditional AES. Because AES' algorithm is strong and employs longer key sizes, it is safer than DES and 3DES. This method also allows for speedier encryption, making it ideal for applications like firewalls and routers that demand either low idleness or high output. Despite the fact that this technology is powerful and has several uses, this method was found vulnerable by using a side-channel attack [11].

2.2 DES

The DES is the most extensively used for encryption technologies since the 1970s [12]. It is a symmetric block encryption technique. Whereas some older applications and systems are still employing DES encryption, despite the fact that DES is now considered insecure for to the short key size used in DES encryption, which can be easily broken by today's current computer platforms. In [13], Ramya G et al. enhanced the DES algorithm's performance by expanding the key length to 128 bits, as opposed to the suggested technique, which inserts keys fully independently.

When this method was created [11], it was a well-designed block cipher that was frequently used. However, this technique was subject to brute force assaults, as it could be cracked via an exhaustive key search. As a result, this method was replaced by AES, a considerably more powerful and efficient algorithm. The KE-DES [14] is an improved version of the DES algorithm developed in this study. A new KD function uses to provide improved data security and efficacy in textual data encryption.

2.3 Blowfish

In 1993 [15], Blowfish was first produced. Blowfish is a symmetric block cipher with a variable key length that is 64 bits long. The algorithm is divided into two sections: a key expansion section and a data encryption section. [16], a comprehensive evaluation was done by the cryptography methods like AES, DES, 3DES, RSA, and Blowfish. Which they stated that the blowfish algorithm is strongest for guessing attacks.

3 Proposed Technique

The suggested Random key generation technology is based on the calculating complexity of the random matrix multiplication and determinant of the same size large matrices to generate a set of keys by using the Matrices Transpose function.

The algorithm goes through each character of the input text and generates an encrypted message only once at a time. The algorithm goes till the entire text encryption is generated as shown in Fig. 1.

Flow Chart of Proposed Method:

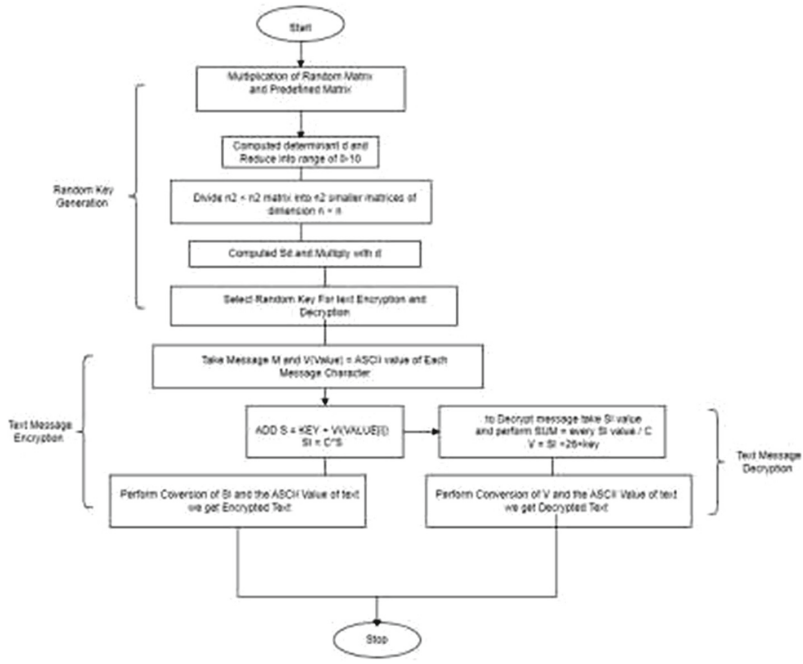

Fig. 1. Flow chart of proposed method

3.1 Random Key Generation Algorithm

1. The First step Generate a 9×9 or $n^2 \times n^2$ random Matrix.
2. Create a predefined matrix size of $n^2 \times n^2$.
3. Compute Multiplication between the generated random matrix and a predefined matrix size of $n^2 \times n^2$.
4. Perform a Matrix Transpose operation of Multiplication matrix.
5. Calculate the determinant d of the matrix generated in the previous step.
6. Reassign it to d in the rand between 0–10.
7. Create a smaller matrix n x n from transpose matrix $n^2 \times n^2$.
8. Compute Determinants of all smaller matrix
9. Perform Multiplication of d with all smaller matrix determinants.
10. Reduce the determinant number in the range of 0–99.
11. Select any Random key from the random numbers, selected primary key we can use for encryption and decryption operations.
12. Reduce the key to the range of 0–10 to generate a multiplication factor k.

The algorithm creates random key generation and it is very complex and hard to perform message decode operations and generate the key by a third person. This algorithm generates every time different matrix, different key and also different encryption for the same message every time as shown in Fig. 2, 3 and 4. Because the method of computing keys requires randomization and sophisticated matrix computations, the adversary will

not be able to get the right key in the time allotted. The technique becomes exceedingly complex and impossible to obtain the key within the entire time provided by internet services since many matrix transformations are necessary and the key generation is random. As a result, this method will be considered safe for its intended use.

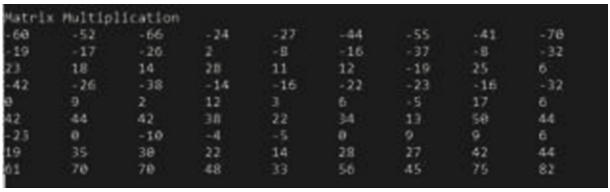

Fig. 2. Random matrix multiplication

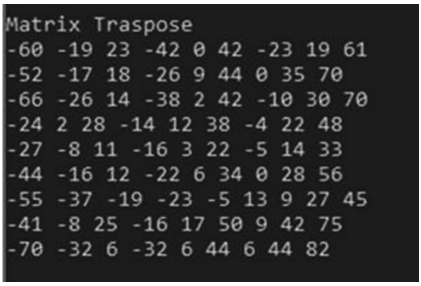

Fig. 3. Perform transpose operation of matrix

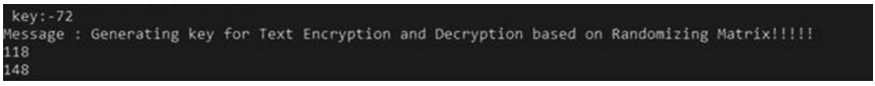

Fig. 4. Selecting a random key for encryption and decryption operations

3.2 Text Message Encryption Algorithm

1. Take text message M for encryption.
2. For every text message character equivalent ASCII value will be denoted as V(Value).
3. Compute Sum S for every ASCII value in the message characters ADD S = KEY + Vi(VALUE[i])
4. Using the previous step calculation we can obtain SI value for every value of the Sum, SI = C * S
   here C is the multiplicative factor computed while generating key
5. For encryption of text messages we can perform the conversion between the value of SI and the ASCII value of text message characters. By performing this conversion our text message is encrypted and it is ready for message communication as shown in Fig. 5.

Fig. 5. Text encryption

3.3 Text Message Decryption Algorithm

1. For decryption of the encrypted text let's take the Si value to be the encrypted message.
2. To find Sum for every SI value by performing a division between every SI value / C (Multiplicative factor).
3. To get the value of V, compute the subtraction of the key from the S value.
4. To obtain the decryption message to perform a Conversion of every value of V to the equivalent value of ASCII character. The generated characters are our decrypted messages as shown in Fig. 6.

Fig. 6. Text decryption

4 Matrix Operations

4.1 Matrix Determinant

*Let, I = n * n matrix*

$$\mathrm{I} = \begin{pmatrix} i11\ i12 \ldots i1n \\ i21\ i22 \ldots i2n \\ \ldots\ \ldots\ \ldots\ \ldots \\ \ldots\ \ldots\ \ldots\ \ldots \\ in1\ in2 \ldots inn \end{pmatrix}$$

formula for the determinant of F(I) $= \sum sgn(\sigma) \prod_{i=1}^{n} i$ *where* $\sigma \in sn$.

4.2 Matrix Multiplication

$$X = \begin{bmatrix} a11\ a12 \ldots a1n \\ a21\ a22 \ldots a2n \\ \ldots\ \ldots\ \ldots\ \ldots \\ \ldots\ \ldots\ \ldots\ \ldots \\ an1\ an2 \ldots ann \end{bmatrix} Y = \begin{bmatrix} b11\ b12 \ldots b1n \\ b21\ b22 \ldots b2n \\ \ldots\ \ldots\ \ldots\ \ldots \\ \ldots\ \ldots\ \ldots\ \ldots \\ bn1\ bn2 \ldots bnn \end{bmatrix} Z = \begin{pmatrix} c11\ c12 \ldots c1n \\ c21\ c22 \ldots c2n \\ \ldots\ \ldots\ \ldots\ \ldots \\ \ldots\ \ldots\ \ldots\ \ldots \\ cn1\ cn2 \ldots cnn \end{pmatrix}$$

where X is n × n matrices and Y is n × n matrices and the result matrices Z is XY.

Such that $\mathrm{C_{ij}} = \mathrm{a_{i1}b_{1j}} + \ldots + \mathrm{a_{in}b_{nj}} = \sum_{k=1}^{m} a_{ik}b_{kj}$, for i = 1 to n and j = 1 to n.

4.3 Matrix Transpose Operation

Transpose of a Matrix can be defined as "A given Matrix which is transform in the form of all rows into columns and vice-versa.

If $A = [a_{mn}]_{p\times q}$, Then $A' = [a_{nm}]_{q\times p}$

$$A = \begin{bmatrix} a11\ a12\ a13 \\ a21\ a22\ a23 \\ a31\ a32\ a33 \end{bmatrix} A^T = \begin{bmatrix} a11\ a21\ a31 \\ a12\ a22\ a32 \\ a13\ a23\ a33 \end{bmatrix}$$

5 Performance Analysis

5.1 Implementation of System

The implementation of proposed algorithm (in C programming language) is performed on a machine that supports the following specifications machine processor with 8 GB RAM Windows 11, 64-bit @ 2.2 GHz Intel i5 10^{th} Generation and to Performing program we use DEV C++ IDE software tool (C Programming Language).

5.2 Text Encryption Algorithms Performance Analysis

When we compute the execution time, throughput, and how much storage is required for execution. of traditional encryption algorithms like BLOWFISH, DES, and AES by using these measurements we can calculate performance analysis. To calculate Throughput perform division between the file size in bytes and the execution time in seconds. High throughput means higher performance because it is required less power consumption.

5.3 Analysis of Complexity for Random Key Generation

In the proposed random key generation algorithm, every step has n number of computations. so, the total time complexity we can say that O(n) of the Random key generation.

5.4 Analysis of Execution Time

In the proposed algorithm computational run time required for first text message encryption and then successfully decoding a specified size text file. In comparison of proposed algorithm and existing algorithms like BLOWFISH, AES and DES using throughput parameter and execution time for algorithm. The comparison is of those system performed using different size of text files which is shows on the Table 1 with their run time. By comparing proposed algorithm is very efficient then the existing algorithm: AES, DES, BLOWFISH are encrypted.

5.5 Analysis of Throughput

Table 1. Analysis of execution time

Comparison between existing algorithms and proposed algorithm					
Input file size	Existing algorithms			Matrix operations	
	DES	AES	BLOWFISH	Converse of matrix	Transpose of matrix
20 KB	2	4	2	7	6.5
40 KB	5	8	4	10.6	9.5
150 KB	20	30	16	12.8	10.2
250 KB	30	44	24	13.4	11.1
Total consummation time	57	86	46	43.8	37.3
Average consummation time	14.25	21.5	11.5	10.95	9.325

Consumption of Power analyze by using throughput. Unit of throughput is Kilobytes per Second (Table 2).

Table 2. Throughput analysis

Algorithms	Throughput (KB/sec)
DES	8.07
AES	5.34
BLOWFISH	10
Converse of matrix	10.5
Transpose of matrix	12.3

Throughput is a ratio of total KB data that has been give as an input divided by total time in second required to compute and execute the code.

Throughput = Total Consummation Time/Total Time Taken For Execution

If Throughput is High than we can say that our algorithm is efficient that other algorithms**.**

6 Conclusion

From the throughput, we can see that the matrix transpose has the highest throughput, so we can say that the proposed encryption algorithm is much more efficient than other encryption algorithms. This paper represents a new Symmetric key encryption algorithm to encrypt and decrypt data in form of files. This algorithm works on two matrix-based

operation that is used to encrypt and decrypt the text from the files. Here we can see that average computation time for BLOWFISH, Converse of matrix and Transpose of matrix are same so in conclusion we can say that all three techniques can be used to encrypt and decrypt data efficiently.

References

1. Alenezi, M., Alabdulrazzaq, H., Mohammad, N.: Symmetric encryption algorithms: review and evaluation study. Int. J. Commun. Netw. Inf. **12** (2020)
2. Mante, P., Harsh, R., Swain, D., Deshpande, D.: A symmetrical encryption technique for text encryption using randomized matrix based key generation **10**(100) (2020)
3. Panda, M., Nag, A.: Plain text encryption using AES, DES, and SALSA20 by java-based bouncy castle API on windows and Linux. In: IEEE 2015 Second International Conference on Advances in Computing and Communication Engineering (ICACCE), Dehradun, India (2015.5.1–2015.5.2), pp. 541–548 (2015). https://doi.org/10.1109/ICACCE.2015.130
4. Dixit, P., Gupta, A.K., Trivedi, M.C., Yadav, V.K.: Traditional and hybrid encryption techniques: a survey. In: Perez, G.M., Mishra, K.K., Tiwari, S., Trivedi, M.C. (eds.) Networking Communication and Data Knowledge Engineering. LNDECT, vol. 4, pp. 239–248. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-4600-1_22
5. Daemen, J., Rijmen, V.: "AES Proposal: Rijndael" (PDF). National Institute of Standards and Technology. p. 1. Archived (PDF) from the original on 5 March 2013, 9 March 2003. Accessed 21 Feb 2013
6. Fadul, I.M.A., Ahmed, T.M.H.: Enhanced security of Rijndael algorithm using two secret keys. Int. J. Secur. Appl. **7**(4), 127–134 (2013)
7. Mehla, R., Kaur, H.: Different reviews and variants of advance encryption standard. Int. J. Sci. Res. (IJSR), 1895–1896 (2012). ISSN (Online): 2319-7064 Impact Factor (2012):3.358
8. Hameed, S., Riaz, F., Moghal, R., Akhtar, G., Ahmed, A., Dar, A.G.: Modified advanced encryption standard for text and images. Comput. Sci. J. **1**(3), 120–129 (2011)
9. Aparna, V.S., Rajan, A., Jairaj, I., Nandita, B., Madhusoodanan, P., Remya, A.A.S.: Implementation of AES algorithm on text and image using MATLAB. In: IEEE 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, (2019.4.23–2019.4.25), pp. 1279–1283 (2019). https://doi.org/10.1109/ICOEI.2019.8862703
10. Thinn, A.A., Thwin, M.M.S.: Modification of AES algorithm by using second key and modified subbytes operation for text encryption. In: Alfred, R., Lim, Y., Ibrahim, A., Anthony, P. (eds.) Computational Science and Technology. LNEE, vol. 481, pp. 435–444. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-2622-6_42
11. Mante, P.G., Oswal, H.R., Swain, D., Deshpande, D.: A symmetrical encryption technique for text encryption using randomized matrix based key generation. In: Borah, S., Emilia Balas, V., Polkowski, Z. (eds.) Advances in Data Science and Management. LNDECT, vol. 37, pp. 137–148. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-0978-0_13
12. Hamza, A., Kumar, B.: A review paper on DES, AES, RSA encryption standards. In: 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART) (2020). https://doi.org/10.1109/smart50582.2020.9336800
13. Bansal, D.R., Thakur, P.: Improved key generation algorithm in data encryption standard (DES) (2016)
14. Reyad, O., Mansour, H.M., Heshmat, M., Zanaty, E.A.: Key-based enhancement of data encryption standard for text security. In: 2021 National Computing Colleges Conference (NCCC) (2021). https://doi.org/10.1109/nccc49330.2021.9428818

15. Mandal, P.C.: Superiority of blowfish algorithm. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **2**(9) (2012). ISSN 2277 128X
16. Patil, P., Narayankar, P., Narayan, D.G., Meena S.M.: A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA, and Blowfish. Procedia Comput. Sci. **78** (2016)