Encryption Algorithm using 2-Dimension Key for Information Security

Sarika Y. Bonde, U. S. Bhadade

***Abstract-** Latest development in wireless technology has found many users and wide applications. As the applicants and users are more, Security of data is a main concern. Wireless networks are very common for both organizations and individuals. The transmission of confidential data like e-mails, banking transactions, credit card details etc. on common transmission media is unsecured. To protect the data during transmission is essential for successful operation of system, which mostly rely on this data. In this paper, proposed an enhanced method for data encryption and decryption which guarantees data confidentiality during its transmission over network. User's data is encrypted before transmission by assigning less number of bits to the plain ASCII text. The key used will consist of all plain ASCII text in random fashion and will be treated as 2-dimension array. In this way, data is transmitting in a secure and efficient manner accomplishing the main goal of Cryptography. 2-dimension array result is compared with Advanced Encryption Standard (AES) algorithm. The use of 2-dimension array will provide security and saves effort of the data to be encrypted*

***Keywords-** Cryptography, Encryption, Decryption, ASCII, AES.*

I. INTRODUCTION

In security of data cryptography plays very important role. The information is hides by using cryptography. In cryptographic terminology, the original message being transmitted is known as plaintext. It is then converted into a coded message using a cryptographic algorithm. This process is called encryption. An encrypted or coded message is known as cipher text, and is converted back into plaintext by the process of decryption. The process of decryption is the same as that for encryption but performed in reverse direction. Fig.1 shows the Encryption and decryption process.

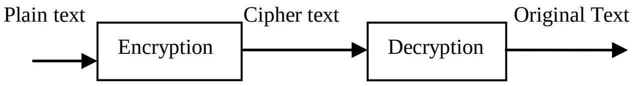

Fig.1: Encryption Decryption Process

Cryptography can essentially be classified into two types, the symmetric and asymmetric type. With a secret or symmetric key algorithm, the key is a shared secret between two communicating parties. Encryption and decryption both use the same key. The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are examples of symmetric key algorithms.

With a public key (PKA) or asymmetric key algorithm, a pair of keys is used. One of the keys, the private key, is kept secret and not shared with anyone. The other key, the public key, is not secret and can be shared with anyone. When data is encrypted by one of the keys, it can only be decrypted and recovered by using the other key. The two keys are mathematically related, but it is virtually impossible to derive the private key from the public key. The RSA algorithm is an example of a public key algorithm [1, 2].

A. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is symmetric-key block cipher and substitution-permutation network which encrypts and decrypts 128bits of the data block. AES uses 128size key, 192 or 256-bit size key which depends on 10 or 12 or 14 rounds. Each processing rounds has 4 steps:

1. Simulate bytes: S-box is used to perform a byte by byte substitution of the block.
2. Rows Shifting: A simple shift operation
3. Mixing column: Each column in row shift is multiplied
4. Add round key: XORed operation with data [1, 2].

B. Organization

This paper has organized into IV sections. Section II presents Literature Survey. Section III is the implementation. Section IV is the experimental results and analysis.

II. LITERATURE SURVEY

Researchers had provided a variety of methods for ASCII based cryptography. Akanksha Mathur et.al [3] has presented an algorithm for data encryption and decryption which is based on ASCII values of characters in the plaintext. The secret used will be modifying another string and that string is used as a key to encrypt or decrypt the data. This algorithm operates when the length of input and the length of key are same. Vineet Sukhraliya et.al [4] has made substitution array using ASCII value for Encryption & Decryption in which randomly generated numbers are used with the help of modulus and remainder by making program in any language i.e. c, c++ and java. After selecting any number randomly use starting and ending number and make subset, followed selection of modulus and remainder as well. Md. Palash Uddin et.al [5] has presented the algorithm based on ASCII conversions and a simple cyclic mathematical function. Also make the encrypted message undoubtedly unprintable using several times of ASCII conversions and a cyclic mathematical function.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Sarika Y. Bonde*, Research Scholar, K.B.C. North Maharashtra University, Jalgaon, (M.S) India. E-mail: sarika_apatil@rediffmail.com

Prof. Dr. U. S. Bhadade, Professor and Head, Department of Information Technology, S. S. B. T. College of Engineering & Technology, Bambhori, Jalgaon, (M.S) India E-mail: umeshbhadade@rediffmail.com

Dividing the original message into packets binary matrices are formed for each packet to produce the unprintable encrypted message through making the ASCII value for each character below 32. Similarly, several ASCII conversions and the inverse cyclic mathematical function are used to decrypt the unprintable encrypted message.

The final encrypted message received from three times of encryption becomes an unprintable text through which the algorithm possesses higher level of security without increasing the size of data or loosing of any data. Hitesh Kumar Sharma et.al [6] has proposed an encryption algorithm which uses the ASCII code to encrypt the plain text. The common key will be used by sender or receiver to encrypt and decrypt the text for secure communication. The algorithms implemented in C# and designed a tool to implement proposed algorithm, also cconcluded that mathematical operations can be applied very easily on ASCII value of the text. Key length should be same as the plaintext length and more execution time these are the limitations of proposed algorithm. A.Vijayan et.al[7] has proposes a new algorithm called AVB algorithm which is used to enhance the security of the data. This algorithm mainly focuses on ASCII value of data. ASCII value of the character is encrypted using normal mathematical calculation for number of time on a particular character and converted to numerical value. Then the cipher text is decrypted to get the original plain text. This algorithm is efficient in two ways it difficult for the intruders to predict the data as each character follows different form of encryption based on the key. Solanki Pattanayak et.al[8] has read a string, then extract each of the single characters from the string and convert these characters to ASCII equivalent value. Apply proposed secret key, along with ASCII value and appropriate encryption algorithm we encrypt the text. Similarly, using the inverse key along with appropriate decryption algorithm we can decrypt the original text. Er. Suraj Arya et.al [9] has proposed ASCII values based technique which uses the string length and some numerical calculation to perform encryption and decryption. To break this technique intruder requires much information about the plain text only single information like string length, , is not sufficient to break this technique. The use of variant string length makes the technique more robust. Further operations apply and depend on the string length. Thus this technique is not depends on any specify key or key generation method it is the strength of the technique. S. G. Rohini et.al [10] has proposed an enhanced algorithm which is based on substitution and shifting techniques. Shifting of the data involves either right shift or right shift. After performing three levels encryption the resultant is again encrypted using shifting technique like left shift operation and right shift operation. The idea of including shifting technique is to make the algorithm more complex, which is more secured and hard to break. Dr. Yaseen Hikmat Ismaiel et.al [11] has uses ASCII to build a coding table in a different way to provide security and saves effort, time and cost. All the above mentioned approaches used ASCII for security of data.

III. IMPLEMENTATION

In general we know that plain ASCII text can be encoded in 7 bits (total 128 characters). To encrypt the characters either we can replace ‘A’ by ‘Z’ or some manipulations can be done on ‘A’ to encrypt it in cipher text. In our method we are encrypting it by assigning less number of bits to the plain ASCII text. The key used will consist of all plain ASCII text in random fashion and will be treated as 2-dimension array. Consider for example, if there are 128 characters in the plain text then every individual character will require 7-bits (2^7 = 128) for encoding it, when 1-dimension key will be used. But if a 2-dimension key is used then we can encode the plain text characters in 5-bits only, thereby saving of 2-bits per character can be achieved.

The detailed explanation is given below:

If all characters are stored in 1-dimension then it will look like

char1, char2, . . ., char126, char127, char128

Now if we store these characters in 2-dimension it will look like:

	Col 1	**Col 2**	**Col 3**	**…**	**Col 31**	**Col 32**
Row0	char1	char2	char3	**…**	char31	char32
Row1	char33	char34	char35	**…**	Char67	char68
Row2	char69	char70	char71	**…**	Char95	char96
Row3	char97	char98	char99	**…**	char127	char128

If the arrangement is done in this way for encoding one character we will need to specify row and column number i.e. 2-bits for row and 5-bits for column with total 7-bits. This seems that the bits are not saved. Now if we repeat most frequent characters w.r.t. English language, the rows will increase from 4 to 8 and now the 2-dimension characters will look like:

	Col 1	**Col 2**	**…**	**Col 17**	**Col 18**	**Col 19**	**…**	**Col 30**	**Col 31**
Row1	char1	char2	**…**	char17	char18	char19	**…**	Char30	Char31
Row2	char1	char2	**…**	char17	Char32	Char33	**…**	Char44	Char45
Row3	char1	char2	**…**	char17	Char46	Char47	**…**	Char58	Char59
Row4	char1	char2	**…**	char17	Char60	Char61	**…**	Char72	Char73
Row5	char1	char2	**…**	char17	Char74	Char75	**…**	Char86	Char87
Row6	char1	char2	**…**	char17	Char88	Char89	**…**	Char100	Char101
Row7	char1	char2	**…**	char17	Char102	Char103	**…**	Char114	Char115
Row8	char1	char2	**…**	char17	Char116	Char117	**…**	Char128	

Thus the above 2-dimension key will of 8*32 where numbers of rows are 8 and columns are 31.

The main advantage of the above structure is as following:

1. For different characters same code will be assigned as the chances of encrypting two consecutive characters in the same row is increased by repeating some characters in each row.
2. We can assign 32 characters using 5-bits, but we are keeping only 31 characters in each row, and will use code ‘11111’ as escape symbol for indicating change in row whenever the two characters to be encrypted will found in different rows.
3. If two consecutive characters are found in same row then only 5-bits code will be used to encrypt the character otherwise first we will put escape symbol followed by new row number and 5-bits code.

IV. EXPERIMENTAL RESULTS & ANALYSIS

The algorithms using 2-dimension predetermine order key and using 2-dimension without predetermine order key successfully executes for different text files size ranges from 1 KB to 200KB using Turbo C7 Simulator. Experimental result of both 2-dimension predetermine order key and 2-dimension without predetermine order key are also compare with AES encryption algorithm.

The size of plain text file after converting to a cipher text is called Encrypted text file size. The size of cipher text after converting to plain text is called Decrypted text file size.

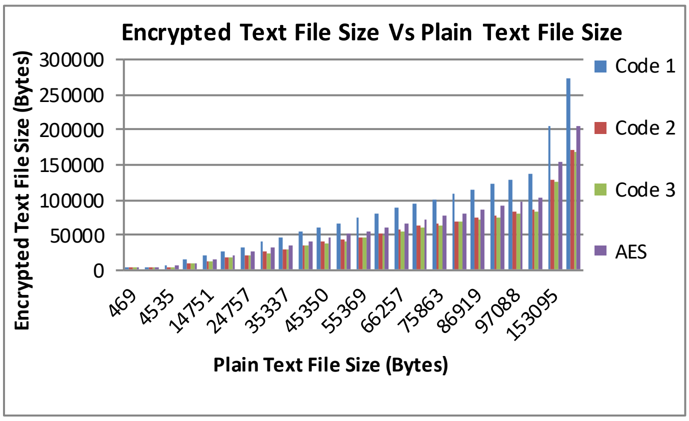

Fig.2: Encrypted Text File Size and Plain Text File Size and comparison for 2-dimension predetermine order key

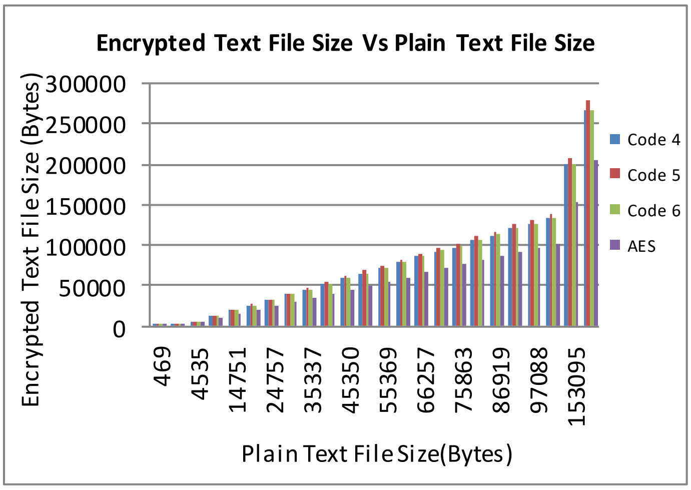

Fig.3: Encrypted Text File Size and Plain Text File Size comparison for 2-dimension without predetermine order key

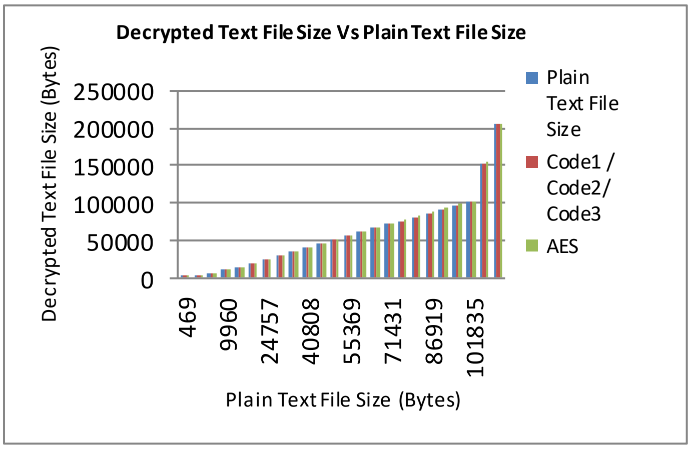

Fig.4: Decrypted Text File Size and Plain Text File Size comparison for 2-dimension predetermine order key

Code1, Code2 and Code3 are the 2- dimension predetermine order key which generated by using 128 random numbers from 0 to 127.

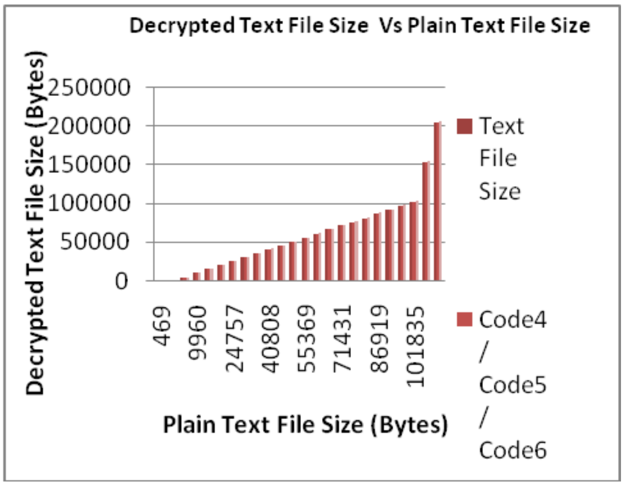

Fig.5: Decrypted Text File Size and Plain Text File Size comparison for 2-dimension without predetermine order key

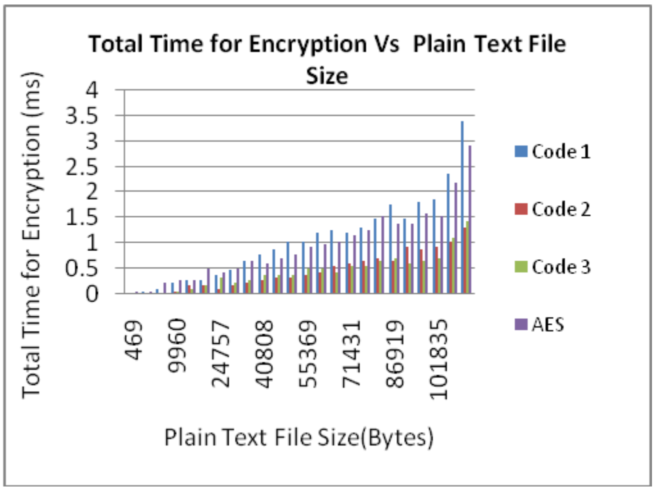

Fig.6: Total Time for Encryption and Plain Text File Size comparison for 2-dimension predetermine order key

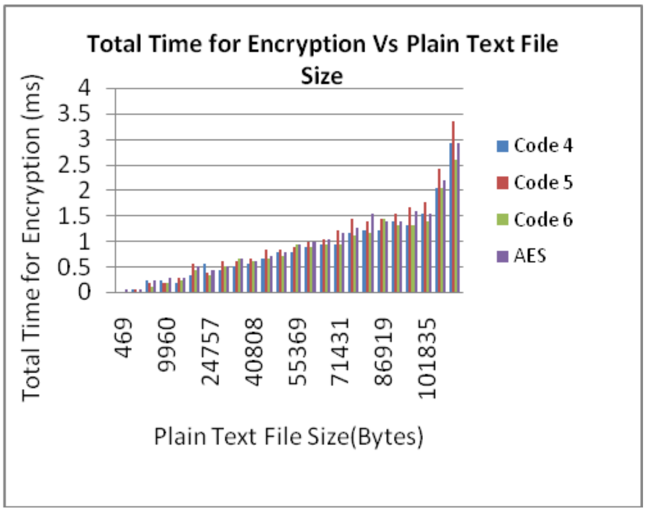

Fig.7: Total Time for Encryption and Plain Text File Size comparison for 2-dimension without predetermine order key

Code 4, Code5 and Code6 are the 2- dimension without predetermine order key which generated by using 128 random numbers from 0 to 127.

Experimental result finding is as follows:

1. From figure 2, Encrypted Text File Size and Plain Text File Size comparison for 2- dimension predetermine order key it is observed that Encrypted Text File Size is less by using Code 3.
2. From figure 3, Encrypted Text File Size and Plain Text File Size comparison for 2- dimension without predetermine order key it is observed that Encrypted Text File Size is less by using Code 6.
3. From figure 4, Decrypted Text File Size and Plain Text File Size comparison for 2- dimension predetermine order key it is observed that by using Code1/2/ 3 Decrypted Text File Size is exactly same as plain text file size , but AES does not give Decrypted Text File Size exactly same as plain text file size .
4. From figure 5, Decrypted Text File Size and Plain Text File Size comparison for 2- dimension without predetermine order key it is observed that by using Code4/5/ 6 Decrypted Text File Size is exactly same as plain text file size , but AES does not give Decrypted Text File Size exactly same as plain text file size .
5. From figure 6, Total time for Encryption and Plain Text File Size comparison for 2- dimension predetermine order key it is observed that Code 3 required least Encryption Time. AES required 1.1% more time for Encryption as compare with code1/2/3.
6. From figure 7, Total time for Encryption and Plain Text File Size comparison for 2- dimension without predetermine order key it is observed that Code 6 required least Encryption Time. AES required 0.1% more time for Encryption as compare with code 4/5/6.

V. CONCLUSION

The key used is 2-dimension array and it consists of all plain ASCII text in random fashion. For performance evaluation 2-dimension array with predetermine order key algorithm and 2-dimension array without predetermine order key algorithm are used. 2-dimension key will of 8*32 where numbers of rows are 8 and columns are 31. We can assign 32 characters using 5-bits, but we are keeping only 31 characters in each row. So by using 2-dimension key number of bit used is reduced. The algorithm successfully executes for different size of text files i.e. 1 KB to 200KB using Turbo C7 Simulator. Based on the experimental result it is concluded that Code 3 is best for Encryption and Decryption using 2-dimension predetermine order key. By using 2-dimension without predetermine order key Code 6 is best for Encryption and Decryption. Further the total time required for encryption is less using code 3 and code 6. Advanced Encryption Standard (AES) algorithm required 0.2% more Encrypted Text File Size as compared with 2- dimension array. AES algorithm required 1.1% more time for Encryption as compared with 2- dimension predetermine order key. AES algorithm required 0.1% more time for Encryption as compared with 2- dimension without predetermine order key. 2- dimension predetermine order key required 1% less time for Encryption as compared with 2- dimension without predetermine order key. So, the use of 2-dimension array will provides security and save effort of data encryption.

REFERENCES

1. Bruce Schneir, “Applied Cryptography”, 2nd edition, John Wiley & Sons, 2007.
2. William Stallings, “Cryptography and Network Security”, Pearson Education, Fourth Edition, 2007.
3. Akanksha Mathur, “A Research paper: An ASCII value based data encryption algorithm and its comparison with other symmetric data encryption algorithms”, International Journal on Computer Science and Engineering (IJCSE), ISSN : 0975-3397, Vol. 4, No. 09, September 2012, pp.1650-1657.
4. Vineet Sukhraliya, Sumit Chaudhary, Sangeeta Solanki, “Encryption and Decryption Algorithm using ASCII values with substitution array Approach”, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 8, August 2013, pp.3094-3097.
5. Md. Palash Uddin, Md. Abu Marjan, Nahid Binte Sadia and Md. Rashedul Islam, “Developing a Cryptographic Algorithm Based on ASCII Conversions and a Cyclic Mathematical Function”, IEEE 3rd International Conference on Informatics, Electronics & Vision, 2014, pp.-1-5.
6. Hitesh Kumar Sharma, Ravi Tomar, J.C. Patni, “HRJ_Encryption: An ASCII Code Based Encryption Algorithm and its Implementation”, IEEE 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015, pp.1024-1027.
7. A.Vijayan, T.Gobinath,M.Saravanakarthikeyan, “ASCII Value Based Encryption System (AVB)”, Int. Journal of Engineering Research and Applications, ISSN: 2248-9622, Vol. 6, Issue 4, (Part - 5) April 2016, pp.08-11.
8. Solanki Pattanayak and Dipankar Dey, “Text Encryption and Decryption With Extended Euclidean Algorithm and Combining The Features of Linear Congruence Generator”, International Journal of Development Research ISSN: 2230-9926,Vol. 06, Issue, 07, July, 2016, pp.8753-8756.
9. Er. Suraj Arya , Dr.Ankit Kumar, “ASCII Based Encryption Decryption Technique for Information Security and Communication”, 3rd International Conference on Innovative Trends in Science, Engineering and Management,(ICITSEM-17),January 2017, pp.-25-33.
10. S. G. Rohini, Ch. Jyothsna, Ch. Ramaiah, Sk. Madeena Sunny, “ASCII Based Symmetric Key Algorithm for Data Security”, International Journal of Pure and Applied Mathematics, Volume- 116 ,No. 5 2017,pp. 75-80.
11. Dr. Yaseen Hikmat Ismaiel, “Coding instead of encryption”, International Journal of Computer Science and Information Security (IJCSIS), Vol. 16, No. 1, January 2018,pp.-1-7.

AUTHORS PROFILE

Sarika Y. Bonde, is Research Scholar from K.B.C. North Maharashtra University Jalgaon, (M.S) INDIA. She has completed her M.E. from Dr.Babasaheb Ambedkar Marathwada University, Aurangabad(M.S.) and B.E. from North Maharashtra University Jalgaon, (M.S).

She is Life Member of ISTE. She has Published/Presented 12 papers in National/International Conference and Journal. Her area of interest is Cryptography, Digital Signal Processing, Object Oriented Programming.

Prof.Dr. U. S. Bhadade, has completed his Ph.D. from M.S. University Baroda, M.E. from Amravati University(M.S) and B.E from Pune University (M.S.). He is Board Member of IJETT (Journal) and Life Member of ISTE, IJERIA and IETE. He has Published/Presented 50 papers in National/International Conference and Journal. His area of interest is Text Compression, Microprocessor, Computer Network, Software Engineering.