ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Data Encryption Using Square Grid Transposition

Bandlamudi.S.B.P. Rani
M.Tech, CST,
Sir C R Reddy College of Engineering,
Eluru-534007, Andhra Pradesh.

Dr. A.Yesubabu
Head of the Department,
Dept of CSE,
Sir C R Reddy College of Engineering,
Eluru-534007, Andhra Pradesh.

M.Krishna
Sr.Assistant Professor,
Dept of CSE,
Sir C R Reddy College of Engineering,
Eluru-534007, Andhra Pradesh.

Abstract:

In the today's world, security is required to transmit confidential information over the network. Security is also demanding in wide range of applications. Cryptographic algorithms play a vital role in providing the data security against malicious attacks This paper has a new approach for encryption of various file formats like text, image, audio and video with key wrapping is proposed. The file is considered as a binary string. The binary data of file is divided into equal sized blocks called as grids. The bit stream of each grid is taken and Square grid transposition is applied. A variable length key based on the grid size considered, say 160 bits for 32-sized grid, 384 bits for 64-sized grid is generated. The key is wrapped up with some public key algorithm say RSA for secret key transposition so that intruder cannot identify. For decryption the reverse grid transposition and private receiver's private key.

Key words:

Cryptography, Grid Transposition, columnar transposition, key wrapping, Session Key.

1. Introduction:

Today the prominence of internet in day to day life has increased a lot. Various transactions in defense, file transfers in an organization internally requires network [6] security. With the availability of internet, many intruders across the world can access our data. In order to rescue our data from intruders we need CRYPTOGRAPHIC [7] techniques. Cryptography [2,6,7,8] is conversion of original data into some modified form of data called cipher. Various algorithms [1,2,3,4,5] have been proposed till now and each has their merits and demerits. As a result researchers are working in the field of cryptography to enhance the security further. In this paper a new approach is proposed where the file is considered as a stream of bits and constructed as various grids.

The technique transforms each grid into encrypted grid by applying bit permutations [3,4,5]. The key generated is also wrapped to get encrypted key. The efficiency of this method is that it supports a variable length grid, secure variable length key. The file can be recovered using the reverse algorithm.

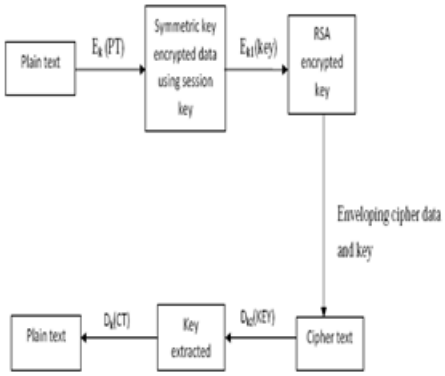
2. Methodology:

Step 1: A square grid of required size is constructed by taking the binary data from source file.
Step 2: Now grid transposition is applied by reading data as various circles starting from the center of the grid to the bottom left at various levels and writing it down on columnar basis from top left to bottom right.
Step 3: A secret key that varies with each session as combination of 0's and 1's is generated based on the grid size, say 160 – bit for 32 – sized, 384 – bit for 64 – sized etc., to produce decimal sequence. Accordingly columnar transposition is done on the grid.
Step 4: A new grid is generated after transposition.
Step 5: The new grid is converted into ASCII sequence and written to another file called encrypted file.
Step 6: Steps 1 to 5 are repeated until the total file is formed into grids and encrypted. Padding with 0's is done in grid formation deficiency.
Step 7: Key generated for each file is encrypted with the public key of sender using RSA Algorithm. This technique of hiding the key is called key wrapping.
Step 8: The encrypted key is then divided into various blocks and appended to file. Hence the new encrypted file with wrapped key is generated.

Detailed Explanation:

The operational structure of the technique is as follows.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal



This technique forms the raw data from the file into various grids of sizes say 32, 64, 128… by reading the binary data as bytes. A grid is nothing but a 2- dimensional array of equal size. The size of the grid is fixed for a file for each session. If the grid is not filled with the data from the file can be pad the grid by adding 0's at the end. The grid thus formed is encrypted starting from the center of the grid to the bottom left. This process is done for all the grids formed.

Now a sequence of bits that varies with grid size as a combination of 0's and 1's is generated. The key generated at each session for a given file is different and size of key varies with the grid size. The security of the data is increased and also the complexity of key generated. Now the session key is encrypted using the RSA algorithm. This makes the key generated secure from intruders. Longer the size of the grid, longer is the size of the key.

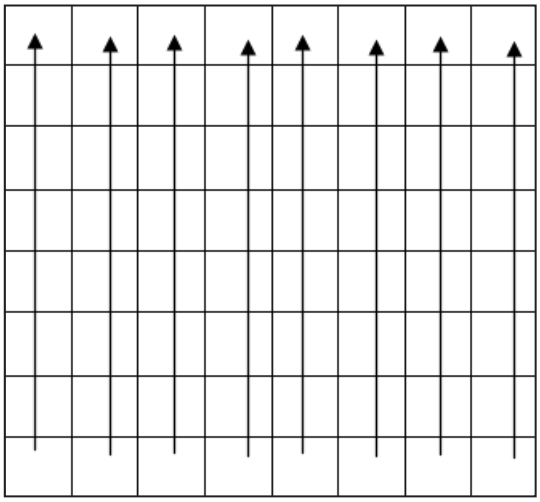
2.1 Encryption:

Broadly our technique can be divided into 3 phases 1) Grid transposition of data. 2) Columnar transposition based on session key. 3) RSA encryption of key.

2.1.1 Grid transposition:

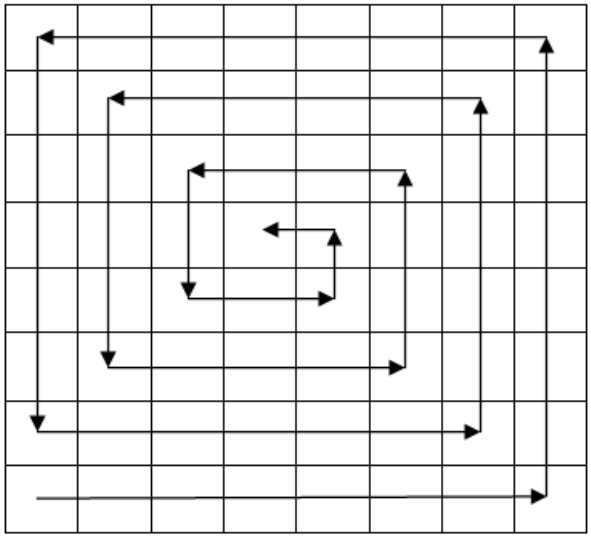
Let us suppose the size of grid is say 8. The grid transposition is as follows: Now 64 bits (8X8) are taken from the file, i.e. 8 bytes are read from the file and is converted to binary string by taking the ASCII values of each byte. This is arranged as a grid starting from top left row by row and ending at bottom right.. Now the data is read starting from the bottom right to top left column by column. The reading can be shown as below.

GRID READING:



The data read as above format is transposed by writing the data into a new equal sized grid starting from bottom left corner to the centre in an helical manner as shown in figure.

GRID WRITTING:



Now the raw data is applied a form of transposition where the intruder cannot identify the hidden data.

2.1.2 Columnar transposition:

This includes the generation of a sequence of 0's and 1's whose size varies with the grid size. The generated sequence is then converted to a decimal sequence by grouping the bits together so that all the non negative indexing for the grid is generated. Discussion of the key generation in detail:

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Suppose if the grid size is equal to 8, then all the non negative sequencing of grid size 8 has to be generated. Each index requires at most 3 bits to represent. Now the size of the key should be 8X3= 24 bits such that by grouping any 3 bits we get the all the indexing and no indexing once occurred should not repeat in the key. Suppose if the grid size is equal to 32, then all the non negative sequencing of grid size 32 has to be generated. Each index requires at most 5 bits to represent. Now the size of the key should be 32X5= 160 bits such that by grouping any 5 bits we get the all the indexing and no indexing once occurred should not repeat in the key.

Suppose the grid size is equal to 64, then all the non negative sequencing of grid size 64 has to be generated. Each index requires at most 6 bits to represent. Now the size of the key should be 32X6= 384 bits such that by grouping any 6 bits we get the all the indexing and no indexing once occurred should not repeat in the key. Accordingly columnar transposition is done based on the session key generated by reordering the elements of each row of grid as per the new key. Now that the grid is converted back to a binary string and is divided into 8 bits each and is written to a new file. The same process is applied for all the grids. Hence the generation of cipher i.e. encryption of data of file is completed.

RSA encryption of key:

The key generated is encrypted based on the public key given to the sender. Instead of encrypting the entire key at a time, it is first divided into N/2 decimal parts where N is grid size. Now each part is considered and is encrypted by using the RSA algorithm. Each part produces 4 bytes of data. Hence a 32 sized grid has an encrypted key written to the file of size 512. Similarly for 64-sized grid it is 1024 and so on. This process of encrypting the key is called encapsulation.

2.1.3 RSA algorithm:

1. Assume any two prime numbers P , Q
2. Calculate N = P * Q
3. Calculate Z= Φ(N) = Φ (P * Q)
= Φ (P) * Φ (Q) (According to
modular arithmetic)
= (P-1) * (Q-1)
4. Assume a value ‘e’ i.e. relatively prime to Z and e < Z and gcd (e , Z) = 1
5. Calculate d, such that e * d 1 (mod Z) 1 (mod Φ(N))
6. Cipher (C) = (m ^ e) mod N
Plaintext (m) = (C ^d) mod N
Thus our key generated is both complex and secure.

2.2 Decryption:

For decryption the reverse process is applied. The individual bytes from the file are combined and the combined result is decrypted using the private key at receiver’s side. Hence the session key is obtained. The reverse process is done i.e. columnar re-transposition and anti grid transposition to get the plain data.

3. Results:

The desired technique has been implemented successfully using JAVA programming language and various files have been experimented with varying file sizes and grid size.

A. With grid size=32:

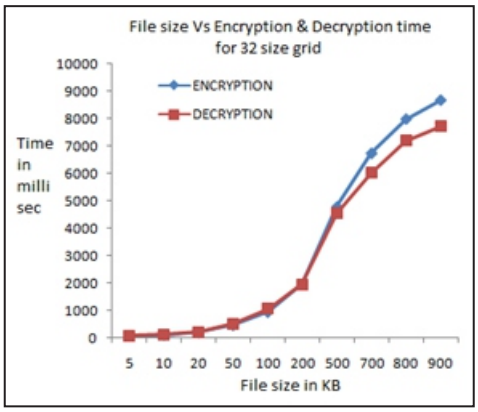
The technique is implemented on .txt files. The same encryption and decryption times continue though it may be applied for other types of file as the algorithm is reading the binary data from file. The results are tabulated as follows.

File size v/s Encryption time for .txt files with 32- sized grid

Source file	File size Before Encryption (in KB)	File size After Encryption (in KB)	Encryption Time (milli Sec)	Decryption Time (milli Sec)
File1.txt	5	5	79	74
File2.txt	10	10	99	117
File3.txt	20	20	191	211
File4.txt	50	50	460	497
File5.txt	100	100	938	1053
File6.txt	200	200	1952	1934
File7.txt	500	500	4774	4535
File8.txt	700	700	6732	6027
File9.txt	800	800	7983	7195
File10.txt	900	900	8672	7712

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The graph that compares the encryption & decryption time is as follows for 32 sized grid.



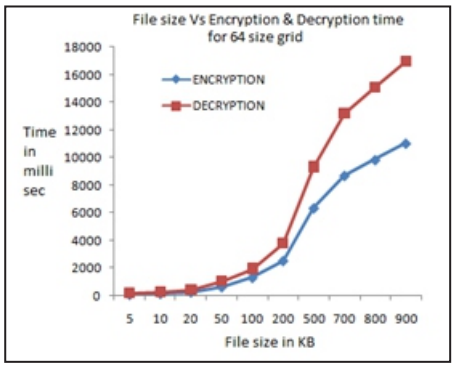
B. With grid size= 64:

With the change of grid size the complexity in key changes and the time taken to form the grid changes hence the encryption and decryption time also increases.

File size v/s Encryption time for .txt files with 64- sized grid

Source file	File size Before Encryption (in KB)	File size After Encryption (in KB)	Encryption Time (milli Sec)	Decryption Time (milli Sec)
File1.txt	5	5	69	203
File2.txt	10	10	138	290
File3.txt	20	20	251	421
File4.txt	50	50	627	1048
File5.txt	100	100	1301	1956
File6.txt	200	200	2521	3805
File7.txt	500	500	6337	9316
File8.txt	700	700	8689	13213
File9.txt	800	800	9834	15092
File10.txt	900	900	11028	16943

The graph that compares the encryption & decryption time is as follows for 64 sized grid.



Let us compare the efficiency of our algorithm with DES algorithm with a block size of 64. the various encryption and decryption times are as follows:

File size v/s Encryption time for .txt files with DES Algorithm with block size=64

Source file	File size Before Encryption (in KB)	File size After Encryption (in KB)	Encryption Time (milli Sec)	Decryption Time (milli Sec)
File1.txt	5	5	452	413
File2.txt	10	10	635	617
File3.txt	20	20	2012	1956
File4.txt	50	50	3845	3468
File5.txt	100	100	7154	7034
File6.txt	200	200	9425	9254
File7.txt	500	500	30654	28546
File8.txt	700	700	39564	37464
File9.txt	800	800	45687	42648
File10.txt	900	900	51367	49264

Thus we can explain that our algorithm is efficient when compared to DES both in case of time taken and key size.

4. Conclusion and Future Enhancement:

The proposed technique has been implemented for text files, image files, audio and video files with variable grid sizes of length 32, 64, and 128. It is being executed efficiently for 32 and 64 but there is a delay with grid size=128 as more time is required to form a big grid of size 128.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Due to some padding (with 0) and enveloping of key the cipher file size varies from the original but not to a greater extent. Since the key is generated randomly with varying size based on the grid size differently for different sessions the complexity of key is enhanced. Due to key wrapping its security is further enhanced. Thus the proposed technique is implemented effectively. Further enhancement of this technique can be the implementation of other asymmetric key cryptographic algorithms in place of RSA.

5. References:

[1] S. Tanmay Bhattacharya , Tamal Kanti Bhattacharya and .R.Bhadra Chaudhuri A General Bit Level Data Encryption Technique using Helical & Session Based Columnar Transpositions. 2009 IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6-7 March 2009.

[2] Jayanta Kumar Pal, "Composite Transposition Substitution Chaining Based Cipher Technique", ADCOM 2008,pp.433-439, 2008

[3] Jayanta Kumar Pal, "A Random Block Length Based Cryptosystem through Multiple Cascaded Permutation-Combinations and Chaining of Blocks", Fourth International Conference on Industrial and Information Systems, ICIIS , Sri Lanka, pp. 26 – 31, 28 - 31 December 2009.

[4] S. Pavithra and Mrs. E. Ramadevi, "Performance Evaluation of Symmetric Algorithms", Journal of Global Research in Computer Science, Vol. 3, No. 8, 2012.

[5] M.R.K. Ariffin, "A New Efficient Asymmetric Cryptosystem based on the Integer Factorization Problem", 2010 Mathematics Subject Classification. 94A60, 68P25, 11D45. Ministry of Higher Education, MALAYSIA, 2012.

[6] I. Tanenbaum, AS.: "Computer Networks" 2nd edition, Prentice Hall, London. 1989.

[7] Stinson, D.R. : "Cryptography Theory and Practice", CRC Press, London, 1995.

[8] Z. Shi and R. B. Lee. Bit permutation instructions for accelerating software cryptography. In Proceedings of the 11th International Conference on Application-Specific Systems, Architectures and Processors, pages 138-148, July 2000.

[9] Z. Shi and R. B. Lee. Sub word sorting with versatile permutation instructions. In Proceedings of the International Conference on Computer Design(ICCD 2002),pages 234-24 1,September 2002.

[10] Z. Shi and R. B. Lee. Implementation complexity of bit permutation instructions. In proceedings of the Asilomar Conference on Signals, Systems and Computers, November 2003.

[11] Network security Essentials Applications and Standards, William Stallings, Pearson Education, New Delhi.