A Block Cipher using Graph Structures and Logical XOR Operation

CH. Suneetha
Assistant Professor in Engineering Mathematics
GITAM University
Visakhapatnam, India

D. Sravan Kumar
Reader in Physics
SVLNS Govt. College
Visakhapatnam, India

A. Chandra sekhar
Professor in Engineering Mathematics
GITAM University
Visakhapatnam, India

ABSTRACT
Cryptography is the science of making and breaking the secret communications. Due to wide spread use of internet in modern times virtually every computer in the world is connected to every other computer. So, there is a threat of hacking and stealing of the message. In general cryptographic primitives are designed to satisfy particular security objectives which may be built from the basic features of confidentiality and authentication. In the present paper a new block cipher encryption is proposed using adjacency matrix of undirected graph structures. In addition logical XOR operation is technically applied to get a good avalanche effect which is one of the desired properties of a good encryption algorithm.

General Terms
Cryptographic protocol, Encryption, Decryption

Keywords
Adjacency matrix, Logical XOR operation encryption, decryption.

1. INTRODUCTION
Definition:- A graph is a triple G = (V,E,ø) where V is a finite set called the vertices of G, E is a finite set called the edges of G and ø is a function with domain E and co-domain $P_2(V)$. The function ø is sometimes called the incidence function of the graph.

Definition Degrees of vertices:- Let G = (V,E,ø) and v ∈ V be a vertex. The degree of v is defined as d(v) to be the number of e ∈ E such that v ∈ ø(e). i.e., e is the incident on v. Suppose $|V|$ = n, let $d_1 d_2 \ldots . d_n$ where $d_1 \le d_2 \ldots \ldots . \le d_n$ be the sequence of the degrees of the vertices of G, sorted by size. This sequence is referred to as the degree sequence of the graph G.

Definition Directed Graph:- A directed graph or digraph is a triple D = (V,E, ø) where V and E are finite sets and ø is a function with domain E and co-domain VxV. We call E the set of edges of the digraph D and call V the set of the vertices of D.

Definition Adjacency Matrix:- The adjacency matrix of an undirected graph is symmetric and therefore has a complete set of real eigen-values and an orthogonal eigen vector basis. The set of eigen values of a graph is the spectrum of the graph. Suppose two directed or undirected graphs G_1 and G_2 are isomorphic then there exists a permutation of matrix P such that

$$PA_1P^{-1} = A_2 .$$

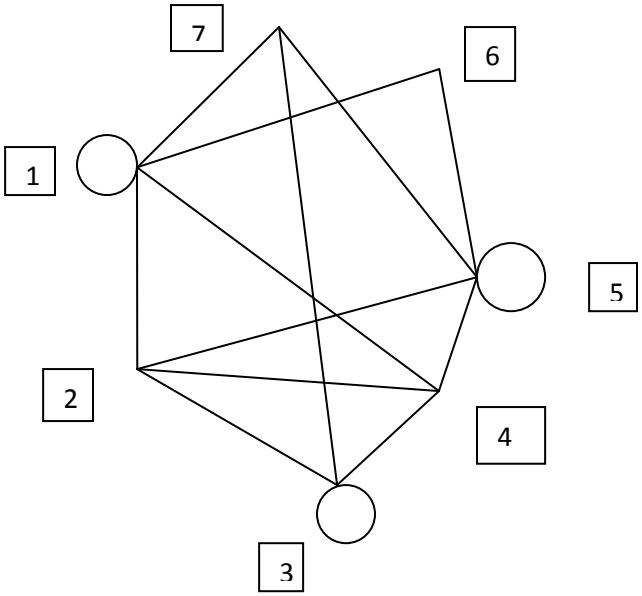
In particular A_1 and A_2 are similar and therefore have the same minimal polynomial, characteristic polynomial, eigen values, determinant and trace. These can therefore serve as isomorphism invariant graphs. However, two graphs may possess the same set of eigen values but not isomorphic. If A is the adjacency matrix of the directed or uni-directed graph G, then the matrix A_n (i.e., the matrix product of n copies of A) has an interesting interpretation: the entry in row i and column j gives the number of (directed or undirected) paths of length n from vertex i to vertex j. The matrix I-A (where I is the nxn unit matrix) is invertible if and only if there are no directed cycles in the graph G. In this case the inverse of (I-A) has the following interpretation: the entry in row i and column j gives the number of directed paths from vertex i to the vertex j (which is always finite if there are no directed cycles). Corresponding to the fact that the number of paths from i to j equals the number of paths of length 0 plus the number of paths of length 1 plus the number of paths of length 2 etc., The main diagonal of every adjacency matrix corresponding to a graph without loops has all zero entries for irregular graphs; d is also an eigen value of A, for the vector and G is connected if and only if the multiplicity of d is 1. An adjacency matrix A determines the graph G. But the opposite is not true. By permuting vertices of G a variety of adjacency matrices can be produced. Hence additional information has to be provided to enforce injective property of the mapping. Several researchers of cryptography [9,10,3] use graph structures and adjacency matrices in their encryption algorithms.

2. XOR CIPHER
An XOR gate is a two input, one output logic circuit, whose output assumes logic 1 state when one and only one of its two inputs assumes logic 1 state. In cryptography XOR Cipher is a simple encryption algorithm but widely used by several researcher of cryptography [6, 7]. A simple repeating XOR cipher is therefore sometimes used for hiding information. Gilbert Sanford Vernam who invented stream cipher and co-invented one-time pad cipher used XOR operation with random or pseudorandom stream of data of the same length to generate the cipher text in his algorithm. The XOR encryption is very easy to crack for professional hackers. Despite its weakness in simple applications, the XOR Encryption remains an important cipher. It is weak when we use repeating keys, but it can be very effective when the key stream varies continuously and

encryption is multiplied. That is why it is often used in modern stream ciphers RC4 (Rivest Cipher #4), developed by Ron Rivest of RSA data security.

2.1 Proposed Method:-

Here the message is encrypted in blocks and each block is encrypted in 7 rounds. In each round of encryption the data is encrypted using the adjacency matrix raised to some power. Between two successive encryption operations using adjacency matrix raised to some power logical XOR is applied. The sender before communicating the messages selects an appropriate graph structure and obtains corresponding adjacency matrix A.



The adjacency matrix corresponding to this graph structure is

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

2.2 Coding of the matrix:-

The adjacency matrix A is a symmetric matrix, hence having all the entries on the main diagonal and all the entries below the main diagonal, one can describe whole matrix and the corresponding graph structure. The entries of the adjacency matrix can be written be written as a sequence $a_{11}a_{21}a_{31}a_{41}a_{51}a_{61}a_{71}a_{22}a_{32}a_{42}a_{52}a_{62}a_{72}a_{33}a_{43}a_{53}a_{63}a_{73}a_{44}a_{54}a_{64}a_{74}a_{55}a_{65}a_{75}a_{66}a_{76}a_{77}$. Hence the equivalent code of the adjacency matrix is n =1101011011100110010100111000. This binary number taking 4 bits at a time can be written as a decimal equivalent number [13061406050308]. This constitutes the first half part of the secret key. The sender also selects the sequence of powers to which the adjacency matrices are raised in each round of encryption which constitutes the second half part of the secret key say [27051734035372]. This sequence of 28 numbers shall be the symmetric secret key for the communication (encryption/ decryption).

2.3 Algorithm:-

Encryption

Step 1: The message is divided into data blocks $D^1,D^2,D^3\ldots\ldots D^n$ of 49 characters each and all the 49 characters of each data block are coded to equivalent decimal numbers using ASCII code table and are written as 7x7 matrices $M^1,M^2,M^3\ldots\ldots M^n$ row-wise. The message always may not contain 49 characters or multiples of 49 characters. In such a case the sender adds three hash characters (###) representing that the message space is over and the remaining characters may be filled at random. He starts encrypting the message from the first data block matrix M^1

Step 2:- M^1 is multiplied with the adjacency matrix A raised to the specific power 27, the first two digits of the second part of the secret key. All the elements are adjusted to mod 255 to get the matrix M^1_{A1}.

$M^1_{A1} = \text{mod}(M^1 * A \wedge 27, 255)$. Then all the elements of M^1_{A1} are converted into equivalent 8 bit binary numbers.

Step 3:- Each element of the matrix M^1_{A1} which is in binary format is XORed with the nearest four neighboring elements to obtain the matrix M^1_{AX1} as follows

$$[M^1{}_{AX1}]_{ij} \rightarrow ((((([M^1{}_{A1}]_{ij} XOR [M^1{}_{A1}]_{ij-1}) XOR [M^1{}_{A1}]_{i+1j}) XOR [M^1{}_{A1}]_{ij+1}) XOR [M^1{}_{A1}]_{i-1j})$$

For example $\left[M^1{}_{A1}\right]_{24} = 10010100$, $\left[M^1{}_{A1}\right]_{23} = 10100010$, $\left[M^1{}_{A1}\right]_{34} = 00111000$, $\left[M^1{}_{A1}\right]_{25} = 10010010$, $\left[M^1{}_{A1}\right]_{14} = 01100101$

then

$$\left[M^1{}_{AX1}\right]_{24} = ((((10010100 XOR 10100010) XOR 00111000) XOR 10010010) XOR 01100101)$$

$$= 11111001$$

Then all the elements of the matrix M^1_{AX1} which are in binary format are converted to equivalent decimal numbers.

Step 4:- The same procedure as described in steps 2 and 3 is repeated in 7 rounds with different powers of the adjacency matrix A

Round 1:- $M^1_{A1} = \text{mod}(M^1 * A \wedge 27, 255)$

All the elements of the matrix M^1_{A1} are converted into binary numbers

$$M^1_{AX1} = XOR(M^1_{A1})$$

All the elements of the matrix M^1_{AX1} are converted to decimal numbers

Round 2:- $M^1_{A2} = \mathrm{mod}(M^1_{AX1} * A \wedge 05, 255)$

All the elements of M^1_{A2} are converted to binary numbers

$$M^1_{AX2} = XOR(M^1_{A2})$$

All the elements of the matrix M^1_{AX2} are converted to decimal numbers

……..
……..
…….

Round 7:- $M^1_{A7} = \mathrm{mod}(M^1_{AX6} * A \wedge 72, 255)$

All the elements of the matrix M^1_{A7} are converted into binary numbers

$$M^1_{AX7} = XOR(M^1_{A7})$$

Then all the elements of M^1_{AX7} which are in 8 binary formats are converted to equivalent text characters using ASCII code table. The same procedure is applied to all the message block matrices M^2,M^3……M^n to obtain the cipher data block matrices $M^1_{AX7}, M^2_{AX7}, M^3_{AX7}.....M^n_{AX7}$. . Then all the elements of these cipher data block matrices are converted into equivalent text characters using ASCII code table to get cipher texts blocks $D^1_E, D^2_E, D^3_E.....D^n_E$. This cipher text is communicated to the receiver in public channel.

Decryption

The receiver after receiving the cipher text starts decrypting the message as follows. The receiver recognizes that the first half part, 13061406050308 of the 28 bit key stream constitutes the entries of the adjacency matrix and the second half part 27051734035372 constitutes the powers to which the adjacency matrix A is raised in each round taking two decimal digits at a time. The first part of the secret key i.e. fourteen decimal numbers are converted to four bit binary numbers taking two numbers at a time. Then all the binary entries are written in the form of a symmetric matrix which is the adjacency matrix A used for encryption/decryption of the message.

Step 1:- The cipher text is divided into data blocks $D^1_E, D^2_E, D^3_E.....D^n_E$ of 49characters each. All the 49 characters of the first data block are converted to equivalent 8 bit binary numbers using ASCII code table and are written as 7x7 cipher data block matrices $M^1_{AX7}, M^2_{AX7}, M^3_{AX7}, ..., M^n_{AX7}$. He starts decrypting from the first data block matrix M^1_{AX7}

Step 2:- He performs logical XOR operation on each element of the matrix M^1_{AX7} with its nearest four neighboring elements to get M^1_{A7} as follows

$$[M^1_{A7}]_{ij} = (((([M^1_{AX7}]_{ij} XOR [M^1_{AX7}]_{i-1\,j}) XOR [M^1_{AX7}]_{ij+1}) XOR [M^1_{AX7}]_{i+1\,j}) XOR [M^1_{AX7}]_{ij-1})$$

Example, $\left[M^1_{AX7}\right]_{45} = 11111001$,

$\left[M^1{}_{AX7}\right]_{35} = 10100010$

$\left[M^1{}_{AX7}\right]_{46} = 00111000$ $\left[M^1{}_{AX7}\right]_{55} = 10010010$

$\left[M^1{}_{AX7}\right]_{44} = 01100101$ then

$$\left[M^1{}_{A7}\right]_{45} = ((((11111001 XOR 01100101) XOR 10010010) XOR 00111000) XOR 10100010)$$

$$= 10010100$$

He converts all the binary numbers of the matrix M^1_{A7} into equivalent decimal numbers

Step 3:- He multiplies M^1_{A7} with the inverse of the adjacency matrix raised to the power 72 and all the elements are adjusted to mod 255 to get the matrix M^1_{AX6} .Then these elements are converted to 8 bit binary numbers.

$$M^1_{AX6} = \mathrm{mod}(M^1_{A7} * inv(A) \wedge 72, 255)$$

Step 4:- The same procedure as described in steps 2 and 3 of the decryption is repeated in 7 rounds to get the matrix M.

Round 1:- $M^1_{A7} = XOR(M^1_{AX7})$

All the elements of the matrix M^1_{A7} are converted to decimal numbers

$$M^1_{AX6} = \mathrm{mod}(M^1_{A7} * inv(A) \wedge 72, 255)$$

All the elements of M^1_{AX6} are converted to binary numbers

Round 2 $M^1_{A6} = XOR(M^1_{AX6})$

All the elements of the matrix M^1_{A6} are converted to decimal numbers

$$M^1_{AX5} = \mathrm{mod}(M^1_{A6} * inv(A) \wedge 53, 255)$$

All the elements of M^1_{AX5} are converted to binary numbers

…….
…….
……

Round 7:-

All the elements of the matrix M^1_{A1} are converted to decimal numbers

$$M^1 = \mathrm{mod}(M^1_{A1} * inv(A) \wedge 27, 255)$$

Similarly the remaining cipher data block matrices $M^2_{AX7}, M^3_{AX7}, ..., M^n_{AX7}$ are decrypted in the same manner to get the original data block matrices $M^1,M^2,M^3……M^n$. Then all the elements of the original data block matrices which are in decimal numbers are converted to text characters to get the original message blocks $D^1,D^2,D^3……D^n$

3. CRYPTANALYSIS AND CONCLUSIONS

In the proposed algorithm the adjacency matrix and the powers to which it is raised in each round of encryption are converted into decimal numbers and are sent to the receiver as a separate communication in public channel. This makes very difficult to retrieve the key matrix by cryptanalysis. Moreover the algorithm proposed here is much secured as long as the key is secret between the communicating parties. Single round of encryption offers inadequate security but multiple rounds of encryption offer increasing security. In the present paper the message is encrypted in 7 rounds with adjacency matrix A raised to different power. Between two successive encryptions using adjacency matrices logical XOR is applied on each element of the matrix with its nearest four neighbouring elements. With this the same characters in the plain text space are mapped to different characters of the cipher text space even though they are in the same text block or different text blocks. So, cipher text is not easily amenable to cryptanalysis [6, 8]. Even the change of a single element of the message matrix changes almost the entire cipher block matrix, i.e., to say that the proposed algorithm has achieved a good avalanche effect [4, 5] which is one of the desired qualities of a good encryption algorithm. If the same message is sent in I and II (or any subsequent) data block, they are mapped to different cipher texts, i.e., even if the same message is sent repeatedly in the same message block, the messages are enciphered to different cipher texts. Hence, active attacks such as chosen plain text attacks [8, 11], chosen cipher text attacks [13] are quite difficult to execute. Hence, the proposed algorithm is less vulnerable to active attacks. The present encryption algorithm is at most secure against man-in-middle attack [2] because the entire master key is agreed upon by the sender and the receiver rather than the electronic exchange of the parts of the key. It is less prone to timing attacks because the time required to encipher or decipher a data block is same for all data blocks since time for enciphering or deciphering is independent of characters in the data block. Even though the original message contains less that 49 characters the remaining characters are filled at random, so that each data block contains exactly 49 characters. Thus the algorithm provides sufficient security against cryptanalysis at relatively low computational overhead.

4. FURTHER SCOPE OF THE WORK

In graph theory graph coloring is a special case of graph labeling. It is an assignment of labels traditionally called colors to the elements of the graph. This coloring of the graphs is widely used in cryptography [14]. Secure multiparty computation allows a group of distributing parties to jointly computes a function of theirs inputs using graphs [1]The communicating parties can share a secret key if they contain an edge of a graph G. The rate of distribution of information as shares can also be obtained by the graphs [1] The cryptographic protocols can be formulated by using Cayley graphs, Expanders and Ramanujan graphs Graph structures can also be used in visual cryptography [12] .

5. REFERENCES

[1]. Amos Beimel, Tal Malkin, Kobbi Nissim and Enav Weinreb, "How should we solve search problems privately", Journal of Cryptology, Springer 2010, Volume-23, Number-2, pages 344-371.

[2]. Anna M. Johnston, Peter S. Gemmell, "Authenticated key exchange Provably Secure against the Man-in-Middle Attack", Journal of Cryptology (2002) Vol. 15 Number 2 pages 139-148.

[3] C. Blundo, A. Santis, D.R. Stinson and U. Vaccaro "Graph decompositions and secret sharing schemes", Journal of Cryptology, 1995, Vol.8, No.1, pgs 39-64.

[4] Carlistle Aams and Stafford Tavares, "The Structured Design of Cryptographically good s-boxes, Journal of Cryptology, 1990, Vol.3, No.1, Pages 27-41.

[5]. Eli Biham, "Cryptanalysis of multiples modes of operation", Journal of Cryptology, 1998, Vol.11, No.1,pgs 45-58.

[6]. Ivan B. Damgard and Lars R, Kundsen, "Two-key Triple Encryption" Journal of Cryptology (1998), Vol. 11, Number 3, pages 209-218.

[7]. John Blaack and Phillip Rogaway "CBC MACs for Arbitrary-Length Messages: The Three-key constructions" Journal of cryptology(2005) Vol. 18 pages 111-131.[8]. J. Kelesey B. Schneir and D. Wagner, "key-schedule cryptanalysis of IDEA,G-DEA,GOST,SAFER and triple-DES. In N. Koblitz editor", Advances in cryptology-Proc.CRYPTO'96, LNCS 1109, pages 237-251.Spro;nger-Verlag Berlin 1996.

[9]. B. Krishna Gandhi, A. Chandra Sekhar and PVDG Prasad Reddy "Cryptographic schemes for digital signals", IETECH International Journal of Advanced Computations, ACo817, 2007.

[10]. B. Krishna Gandhi, A. Chandra Sekhar and S. Srilakshmi "Cryptographic schesmes for digital signals using finite state machines", International Journal of Computer Applications, Vol.29, No.6,Sept-2011.

[11]. Lorenz Minder and Alistair Sinclair, "The Extended k-tree Algorithm" Journal of cryptologyDOI: 10.1007/s00145-011- 9097-y.

[12]. Steve Lu, Dameil Machala and Rafail Osterousky, "Visual Cryptography using graphs", J. Comb Optim, 21:47-66, DOI 10.1007?s 10878-009-9241-x.

[13]. Victor Shoup and Rosario Gennaro "Securing Threshold Cryptosystems against Chosen Cipher text Attack, Journal of Cryptology (2002) Vol15,Number2 pages75-96.

[14]. Yvo Desmedt, Josef Piprzyk, Ron Sternfield, Xiaoming Sun and Tartary, "Graph Coloring Applied to Secure Computation in Non-Abilian Groups", Journal of Cryptology, Springer, online from 5th September 2011.