Stochastic Modelling and Computational Sciences
Vol. 1 No. 1 (June, 2021)

Receiced: 06th January 2021 *Revised:20th March 2021* *Accepted: 08th May 2021*

HYBRID CRYPTOSYSTEM USING GRAPH CIPHERS

M.R. Pathak, faculty of Mathematics, Srimathi Indira Gandhi College, Trichy, India

Abstract

In this paper Hamilton cycles in graphs have been discussed as a tool for encryption/decryption. Identifying a particular Hamilton cycle in a complete graph caters the security of this system. Algorithms for key generation and encryption/decryption process are also discussed. This paper concludes with introducing a hybrid cryptosystem that uses the proposed symmetric key algorithm and RSA algorithm. A comparison of this hybrid cryptosystem with RSA cryptosystem is also presented.

1. INTRODUCTION

Cryptography has been in use from antiquity and since the introduction of public key cryptosystems like RSA [7] and ECC [3], [6] it has attracted the attention of the research community and has become a field in which research is fostering currently. It is the study of mathematical techniques related to information security, with confidentiality, data integrity, authentication, and non-repudiation as primary goals. It branches into two heads depending on the key(s) used namely symmetric key and asymmetric key cryptosystems. Though asymmetric key cryptosystems have gained fame, symmetric key cryptosystems like DES [1], [2], [5] also still persist because they are comparatively faster.

Problems of large computational complexity are available in Graph Theory. Set covering, subset-sum, Traveling Salesman, vertex covering, weighted set covering, Hamiltonian cycle and maximum clique problems are some known NP complete problems in Graph Theory. These problems form a good base for the design of efficient and secure ciphers, more secure than those proposed in the past.

Treating the vertices of a graph as messages and arcs of certain length as tools for encryption, a method was proposed by Vasiliy A. Ustimenko in [8]. In [4] Kamil Kulesza and Zbigniew Kotulski discuss the use of graph coloring as a private key

Key words: *Hamilton cycle, NP-Complete, Symmetric key cryptosystem.*

and the graph structure as public key. This paper discusses the use of Hamilton cycle as a tool for encryption/decryption. Algorithms for key generation, encryption/ decryption is also provided.

The notations to be used in this paper are reviewed first, following which the idea of encryption/decryption using Hamilton Cycles in graphs is introduced. Then security analysis of the proposed system is given. A hybrid cryptosystem using RSA algorithm and the proposed algorithm is given. A comparison of the time complexities of the hybrid system and RSA algorithm is presented.

2. SYMMETRIC KEY CRYPTOSYSTEM USING GRAPHS

Notations

p	Prime number
GF(p)	Galois field generated by p {0,1,2,…,p-1}
G	Complete graph drawn using elements of GF(p)-{0} as vertices
L(G)	Label set of G $\{v_1, v_2, v_3, \ldots, v_{p-1}\}$
k	An arbitrary element from GF(p) whose label is the start vertex
$\{v_{k_j}\}$	Labeled vertex sequence corresponding to the Cayley's table (with multiplication operation) row entry for k
$w_{\max}$	The maximum of the edge weights of G
s_v	Start vertex from which walks on the graph starts for encryption/decryption

$m_j \quad 1 \le j \le n ; n \in Z$ is the sequence of messages to be encrypted

$I_{m_j} \quad 1 \le j \le n ; n \in Z$ is the integer equivalent of the message sequence

$c_j \quad 1 \le j \le n ; n \in Z$ is the encrypted cipher text sequence

2.1. Key Generation

Key generation is an important module in any cryptosystem. The following procedure is used for key generation. A prime number p is chosen at random using which its corresponding prime field excluding zero is generated. The elements of this field serves as vertices for the complete graph G . These elements are relabeled using the label set L(G). Then a number is randomly chosen from GF(p)-{0}.Its Cayley's table row entry is computed with which the Hamilton cycle is modeled. This Hamilton cycle serves as the common secret key.

1. Select a prime p
2. Draw a complete graph with elements of GF(p)-{0} as vertices
3. Label the vertices in some random fashion
4. Attach weights (w 's) to all the edges of G in some random fashion such that there is no replication of weights and construct a cost adjacency matrix for G
5. Find the maximum ($w_{\max}$) of the edge weights from the cost adjacency matrix
6. Choose an integer W> $w_{\max}$ which will act as the base number in the computation of edge sum
7. Choose arbitrarily an element k from GF(p)
8. Compute its corresponding Cayley's table row entry
9. Append k to the end of the sequence obtained
10. Map these elements to their corresponding labels to get the Hamilton cycle.
11. Compute the edge sum of the Hamilton cycle using the formula,

$$\text{sum}= \sum_{i=0}^{P-2} \left(w * W^{i} \right)$$

12. Shared secret key is (W, sum)

The time complexity of the above algorithm is $O(n^2)$ where n= p-1. The Hamilton cycle need to be retrieved before encryption/decryption which is done by performing modulo division over sum. Then encryption/decryption process is geared up using the following algorithm.

2.2. Encryption

Encryption is done with the algorithm given below. The starting vertex of the Hamilton cycle is fixed as the initial start vertex. From the start vertex the data encryption is done by walking along the Hamilton cycle with the numeric equivalent of every letter. Once the number of steps becomes equal to the numerical value then the corresponding vertex is the equivalent cipher. The start vertex is now positioned from this point and encryption proceeds further in a similar manner.

Start

$s_v = v_{k_1}$

For j = 1 to n

For I = 1 to I_{m_j}

c_j = (s_v +1) mod (p-1)

s_v = c_j

Next i

Next j

End

Cipher text corresponding to $\{m_j\}$ is c_j

The time complexity for encryption process is $O(n \ell)$ where n = p-1 and ℓ is the length of the message to be encrypted.

2.3. Decryption

The algorithm given under is used for decryption process. The common secret key, the Hamilton cycle is used for decryption. The start vertex and path length are initialized to the starting vertex of the Hamilton cycle and to zero respectively. Then from the start vertex walks are made along the Hamilton cycle till the cipher text is found. The number of edges passed will give the path length and converting it to alphabet equivalent the plain text is obtained. The start vertex is then changed to the last cipher that was found. Repeating the previous steps decryption proceeds with the cipher till all the encrypted data are decrypted.

Start

$s_v = v_{k_1}$

For j=1 to n

Path_length=0

For i= s_v to c_j

Path_length = Path_length +1

Next i

$s_v = c_j$

I_{m_j} = Path_length

Next j

End $\{I_{m_j}\}$ corresponding to $\{c_j\}$ is obtained using which $\{m_j\}$ can be found

The time complexity for decryption process is $O(n\ell)$ where n = p-1 and ℓ is the length of the message to be decrypted.

2.4. Example

2.4.1 Key Generation

Let p =7. Then GF(p)={1,2,3,4,5,6}. The complete graph G is as shown in Fig. 1

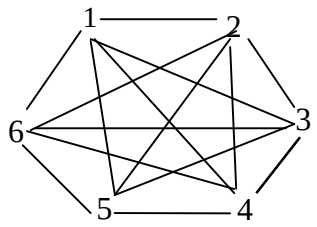


Fig. 1

After a random labeling the graph becomes as shown in Fig. 2

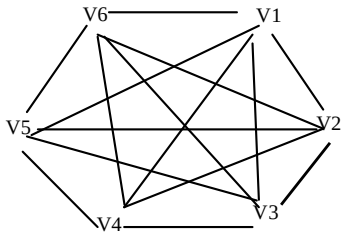


Fig. 2

The cost adjacency matrix after assigning weights is

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	∞	2	7	13	17	19
v_2	2	∞	6	10	11	15
v_3	7	6	∞	3	5	9
v_4	13	10	3	∞	4	16
v_5	17	11	5	4	∞	21
v_6	19	15	9	16	21	∞

w_{max} = 21 from the cost adjacency matrix. Let W= 23 > w_{max}. Arbitrarily choose any integer k= 3 from GF (7). The Cayley table row corresponding to 3 is

3 6 2 5 1 4

Appending 3 to this row, we get

3 6 2 5 1 4 3

Using the labels of vertices of G, the sequence is

v_2 v_5 v_1 v_4 v_6 v_3 v_2

Calculate the edge sum

$$\text{sum} = 11 * 23^0 + 17 * 23^1 + 13 * 23^2 + 16 * 23^3 + 9 * 23^4 + 6 * 23^5 = 41338578$$

Shared secret key is (23, 41338578)

Converting 41338578 to base 23 gives the sequence {11, 17, 13, 16, 9, 6}

Using this sequence the Hamilton cycle is obtained from the cost adjacency matrix as

{v_2 v_5 v_1 v_4 v_6 v_3 v_2}

2.4.2 Encryption

Message : bee

Integer equivalent : 2 5 5 (a = 1, b = 2, ..., z = 26)

Hamilton cycle : v_2 v_5 v_1 v_4 v_6 v_3 v_2

Sv = v_2

I_{m_1} = 2

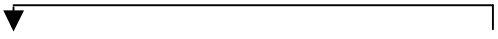
$v_2 \rightarrow v_5 \rightarrow v_1$ v_4 v_6 v_3

$c_1 = v_1$ The cipher text corresponding to ‘b’ is v_1

For ‘e’ ,

I_{m_2} = 5

s_v = v_1



$v_2 \rightarrow v_5$ $v_1 \rightarrow v_4 \rightarrow v_6 \rightarrow v_3$

$c_2 = v_5$. The cipher text corresponding to ‘e’ is v_5

For ‘e’,

I_{m_3} = 5

c_3 = v_2. The cipher text corresponding to 'e' is v_2

The cipher text corresponding to 'bee' is v_1 v_5 v_2

2.4.3 Decryption

Cipher text: v_1 v_5 v_2

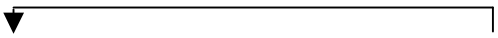
Hamilton cycle : v_2 v_5 v_1 v_4 v_6 v_3 v_2

$c_1 = v_6$

$v_2 \rightarrow v_5$ $v_1 \rightarrow v_4 \rightarrow v_6 \rightarrow v_3$

Hence I_{m_1} = 2

c_2 = v_5



$v_2 \rightarrow v_5$ $v_1 \rightarrow v_4 \rightarrow v_6 \rightarrow v_3$

I_{m_2} = 5

$c_3 = v_2$

I_{m_3} = 5

The decrypted plain text corresponding to v_1 v_5 v_2 is 'bee'

While choosing the value of p it has to be chosen such that it is greater than the maximum of the integer equivalent of the plain text.

3. SECURITY ANALYSIS

The data encryption method provided with the proposed system preserves the cryptographic primitives such as authentication, confidentiality and data integrity.

In the proposed encryption algorithm, the start vertex is changed after encrypting each character. The cipher text c_j is a function of c_{j-1} which increases the confusion in the cipher text. Hence cryptanalysis using statistical tests is curbed.

Also this system is liable only for a brute force attack unlike DES [1], [2] which is subject to differential cryptanalysis (DC), linear cryptanalysis (LC), and Davies' attack**.** The only possible way to recover the key is by brute force attack. The system uses a complete graph and hence an attacker has to try n! possibilities in the worst case and thus landing in a time complexity that is greater than exponential time complexity.

4. HYBRID CRYPTOSYSTEM USING GRAPH CIPHERS

In the above symmetric cryptosystem, key distribution is a problem as with any symmetric key system. Using RSA, to share the key, the system is converted to a hybrid system. This system uses the proposed symmetric key encryption algorithm using graphs for data encryption and RSA algorithm for key encryption. With RSA algorithm encryption operations take $\ell O(k^2)$ steps whereas with the proposed system $O(n\ \ell)$ where n =p-1, k is the number of bits in the modulus and ℓ is the length of the message to be encrypted. A comparison of the Hybrid system with RSA cryptosystem using apriori analysis for time complexity is shown in Fig. 3. The hybrid cryptosystem shows a comparatively better performance as the message size increases.

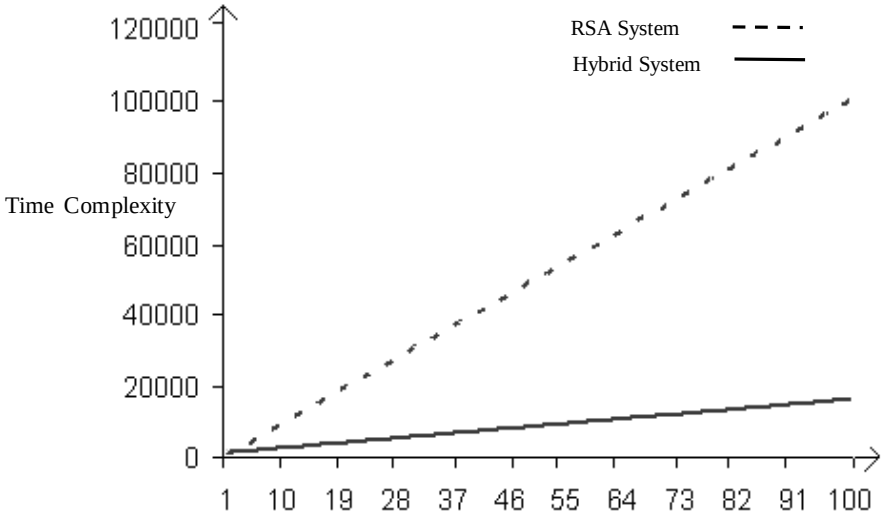


Fig. 3

5. CONCLUSION

A symmetric key cryptosystem using Hamilton cycles in graphs was introduced in this paper. Using that a hybrid cryptosystem was given whose execution time complexity was compared with RSA cryptosystem. Further research is to exploit the NP-complete problems in graphs to develop asymmetric key cryptosystems around them.

REFERENCES

[1] D. Davies and S. Murphy, "Pairs and Triplets of DES S-Boxes", *Journal of Cryptology*, **8**, No. 1, pp. 1-25, 1995.

[2] Eli Biham and Adi Shamir, "Differential cryptanalysis of DES like cryptosystems", *Journal of Cryptology*, **4**, No. 1 pp. 3-72, 1991.

[3] Harald Baier, "Elliptic curves of prime order over Optimal Extension Fields for use in Cryptography", INDOCRYPT 2001, LCNS 2247, 2001.

[4] Kamil Kulesza and Zbigniew Kotulski, "Secret Sharing for n-Colorable Graphs with Application to Public Key Cryptography", *Proceedings of 5th NATO Conference on Military Communication and Information Systems 2003, Capturing New CIS Technologies*,2003 .

[5] National Bureau of Standards, Data Encryption Standard, U.S. Department of Commerce, *FIPS pub*. 46-3, Oct 1999.

[6] Neal Koblitz, Alfred Menezes and Scott Vanstone, "The State of Elliptic Curve Cryptography", *Designs, Codes and Cryptography*, **19**, pp. 173–193, Mar 2000.

[7] Ronald L. Rivest , Adi Shamir and Leonard M. Adleman , "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, **21**, pp. 120-126, Feb. 1978.

[8] Vasiliy A. Ustimenko, "Graphs with Special Arcs and Cryptography", *Acta Applicandae Mathematicae*, **74**, pp. 117-153, Nov. 2002.