Multiple Message Encryption using Euler Graphs
with Hamiltonian Circuit as Key

M. Yamuna
SAS, VIT University
Vellore, Tamil Nadu, India
myamuna@vit.ac.in

Anindya Kumar Das
SITE, VIT University
Vellore, Tamil Nadu, India
anindyascc227@gmail.com

Abstract
The encyclopedic applications of Graph Theory in varied fields of study have additionally created its significant impact in Cryptology. Graphs are extensively used as ciphers for secure communication in the presence of third parties referred to as adversaries. In this paper, we propose an algorithm for deciding if a graph is Euler. We also propose an efficient method of encrypting any text message using Euler graph with a Hamiltonian circuit as a key for encryption.

Keywords- Euler graph, encryption, decryption, incidence matrix.

1. Introduction
Cryptography, fundamentally is the study of methods by which we can send a message in secret (namely, in disguised or in encrypted form) so that only the intended recipient can unravel the disguise and interpret the message (or decipher it). Various algorithms in cryptography have been widely developed by adopting the varied properties of Graph Theory, which are being used nowadays to communicate sophisticated data across domains. Several methods are developed for message encryption using graph theory. In [1] text encryption using graph domination is provided. In [2] encryption using Just excellent graphs is discussed. In this paper we propose an encryption method using Euler graphs.

2. Materials and methods
In this section we will provide the basic properties that are required in support of our propositions.

2.1 Graph
In mathematics and computer science, graph theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects. A "graph" in this context is made up of "vertices" or "nodes" and lines called edges that connect them. A graph may be undirected, meaning that there is no distinction between the two vertices associated with each edge, or its edges may be directed from one vertex to another; see graph (mathematics) for more detailed definitions and for other variations in the types of graph that are commonly considered. Graphs are one of the prime objects of study in discrete mathematics [3].

2.2 Euler Graph
In graph theory, a Eulerian trail (or Eulerian path) is a trail in a graph which visits every edge exactly once. Similarly, an Eulerian circuit or Eulerian cycle is an Eulerian trail which starts and ends on the same vertex. The term Eulerian graph has two common meanings in graph theory. One meaning is a graph with an Eulerian circuit, and the other is a graph with every vertex of even degree [4].

2.3 Hamiltonian Graph
In the mathematical field of graph theory, a Hamiltonian path (or traceable path) is a path in an undirected or directed graph that visits each vertex exactly once. A Hamiltonian cycle (or Hamiltonian circuit) is a Hamiltonian path that is a cycle [5].

2.4 Incidence Matrix
In mathematics and computer science, an adjacency matrix is a means of representing which vertices (or nodes) of a graph are adjacent to which other vertices. Specifically, the adjacency matrix of a finite graph \(G\) on \(n\) vertices is the \(n \times n\) matrix where the non-diagonal entry \(a_{ij}\) is the number of edges from vertex \(i\) to vertex \(j\), and the diagonal entry \(a_{ii}\), depending on the convention, is either once or twice the number of edges (loops) from vertex \(i\) to itself. Undirected graphs often use the latter convention of counting loops twice, whereas directed graphs typically use the former convention [6].
Snapshot 1 provides Graphs and their corresponding incidence matrices[7].
3. Proposed algorithm

- Let v be the number of vertices in the graph G and e be the number of edges.
- Construct an incidence matrix $A[i,j]$ depending upon the said rules.
- Count the number of non-zero elements in each row i.e. $A[i,j]=1$.
- If the count is even for every row, then G is Euler.
- If the count $A[i,j]=1$ is odd for any one of the rows, then G is not an Euler graph.
3.1 Example

Consider the graph $G$ as shown in Fig 2. Here we are going to check whether this graph $G$ is an Euler Graph or not. So, first we need to generate the incidence matrix. We make use of a code to form the incidence matrix as well as check for the said property of an Euler graph. If the property holds, then $G$ is an euler else not.
The snapshot 2 of the output screen shows that the above graph is an Euler graph.

**4. Proposed Encryption Scheme**

The encryption scheme that is being discussed here will encrypt a message into an Euler Graph, which will be later decrypted based on a specific Hamiltonian circuit, traced out from the graph.
This Hamiltonian circuit serves as a key to decrypt the original message from the encrypted graph. We know that a complete graph with \( n \) number of vertices has \( \frac{n-1}{2} \) edge disjoint Hamiltonian circuits, where \( n \) is odd number \( \geq 3 \) [8].

Hence the Hamiltonian circuits will aid as a carrier for the encrypted message such that we will get back the original message only when we decrypt the graph; by traversing through that one specific Hamiltonian circuit.

Each and every letter or alphabet in the message is converted into a stream of binary numbers (0s and 1s) using the encryption technique, as stated below. An adjacency matrix will be constructed based on the binary stream and sent to the receiver.

### 4.1 Encoding Technique

The encoding chart is designed as per our need and interest.

- Every alphabet is first converted into uppercase whose ASCII values are taken into consideration. ASCII values of uppercase characters ranges from 65 to 90.
- These ASCII values are then transformed into binary, by using a decimal to binary base converter. This process generates a binary equivalent of every alphabet in the message.
- Now, the binary equivalent of the alphabet is XORed with the binary equivalent of 32. The symbol \( \oplus \) is used to represent the XOR function.

This step is required because there is a possibility that the number of 1s in the binary equivalent of the alphabet is \( \leq 2 \). In that case, we cannot form an euler graph with just two vertices. We, therefore need at least 3 vertices to form a circuit. So, in order to make sure that we have more than 2 vertices to represent every alphabet as an Euler graph, we use the XOR method. Also we use the number 32 because on XORing 32 with the binary number, the resulting number will never lie in the range of 65-90, thereby not affecting the original alphabet.

For example consider the alphabet ‘A’ whose ascii value is 65. The binary equivalent of 65 is 1000001. Since, there are only two 1s, we cannot form a circuit. So, the need of XORing with 32 arises in such cases. After XORing with 32, the binary stream becomes:

\[
\begin{align*}
1000001 \\
\oplus 0100000 \\
1100001
\end{align*}
\]

yielding three 1s, which is sufficient enough to form a circuit. The decimal equivalent of the result 1100001 is 97, which lies beyond the range of uppercase ASCII values. Hence the alphabet ‘A’ won’t be mapped to some other alphabet, but only ‘A’. 
4.2 Encryption Algorithm

- Convert each character in the message to a binary stream using the encryption technique and store it in an array $A[i]$.
- Count the number of 1s in $A[i]$ i.e count($A[i]=1$).
- Create an adjacency matrix $M[i,j]$ using count($A[i]=1$) which represents the number of vertices.
- For every $M[i,j]$ where $i = j$, make $M[i,j] = 0$ (i.e. simple graph).
- For every $A[i] = 1 \land A[i+1] \neq 0$, make $M[i,j] = 1$ as well as $M[j,i] = 1$ (since adjacency matrix is symmetric).
- For every $A[i] = 1 \land A[i+1] = 0$, count the number of zeros following the 1 and make $M[i,j]$ as well as $M[j,i] = (\text{Number of 0's} + 1$.
  
  i.e if $A[i] = 10$, then $M[i,j]=M[j,i] = 2$.
  
  if $A[...]=1000$, then $M[i,j]=M[j,i] = 4$.
- Repeat the above steps until the tracing of the Hamiltonian circuit reaches the end vertex.
- If the last element of $A[i] = 1$ i.e. the stream ends with 1, then make then $M[i,0]=M[0,j] = 1$.
- If the last element of $A[i] \neq 1$ i.e. the stream ends with 1 followed by a series of 0’s, then make then $M[i,0]=M[0,j] = (\text{number of 0’s}) + 1$.
- Send the adjacency matrix to the receiver.

4.3 Decryption Algorithm

- Receive the adjacency matrix.
- Since the matrix is symmetric, we just need to track either the upper triangular matrix or the lower one, along the main diagonal.
- Create a temporary array $T[i]$ and store the elements from the adjacency matrix, traversing one vertex after another, as we trace the hamiltoninan circuit. i.e. $T[i] = M[i,j]$. 
The contents of \( T[i] \) are expanded to get the binary stream.

Apply the operations in reverse order to get back the original data, using the encryption chart.

**Fig. 3** The proposed Encryption scheme is depicted in the above block diagram.

### 4.4 Example

Let the message to be sent be “MOCK”. Using the encoding chart, we convert each of the alphabets into binary strings.

\[
\begin{align*}
M &\rightarrow 1101101 \\
O &\rightarrow 1101111 \\
C &\rightarrow 1100011 \\
K &\rightarrow 1101011
\end{align*}
\]

An adjacency matrix representing a graph, is generated for each of the alphabets based on the encryption algorithm. The key Hamiltonian circuit that is to be traced out from the corresponding alphabet’s graph is decided to be in an increasing order of vertex numberings i.e \(<1,2,3,4,\ldots>\)
The graph representation along with the adjacency matrix of each of the alphabets are shown in Fig. 4. The matrices corresponding to these graphs are also provided.

![Graphs for M, O, C, K](image)

**Fig. 4** Euler Graphs representation for each of the characters in the message.

The adjacency matrices are send to the receiver one after another with the key Hamiltonian circuit traversal series. When the receiver receives the adjacency matrices, he uses the key to decode the data from these adjacency matrices.

\[
M = \begin{pmatrix}
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 2 & 0 & 0 \\
0 & 2 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 2 \\
1 & 0 & 0 & 2 & 0 \\
\end{pmatrix} \quad O = \begin{pmatrix}
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 2 & 0 & 0 \\
0 & 2 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

\[
C = \begin{pmatrix}
0 & 1 & 0 & 1 \\
1 & 0 & 4 & 0 \\
0 & 4 & 0 & 1 \\
1 & 0 & 1 & 0 \\
\end{pmatrix} \quad K = \begin{pmatrix}
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 2 & 0 & 0 \\
0 & 2 & 0 & 2 & 0 \\
0 & 0 & 2 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 \\
\end{pmatrix}
\]

These adjacency matrices are send to the receiver one after another with the key Hamiltonian circuit traversal series. When the receiver receives the adjacency matrices, he uses the key to decode the data from these adjacency matrices.
5. Implementation of the Algorithm

A program is written in C language to help us to generate these adjacency matrices one after another. The snapshot 3 of the output screen is shown below.

![Snapshot 3](image)

**Snapshot 3** The output screen for encryption.

Using the decryption algorithm, the receiver decodes the following array of elements from each of the adjacency matrices. The same program is used to decrypt the encoded messages. The snapshot 4 of the output screen is shown below.
For the 1st matrix, he receives 12121. 2nd matrix yields 121111. 1411 and 12211 are received from the 3rd and 4th matrices respectively.

On expansion of the each of the array elements he gets,

12121 \rightarrow 1101101
121111 \rightarrow 1101111
1411 \rightarrow 1100011
12211 \rightarrow 1101011

So, on applying the operations in the encoding chart in reverse order,

1101101 produces the letter \rightarrow ‘M’
1101111 produces the letter \rightarrow ‘O’
1100011 produces the letter \rightarrow ‘C’
1101011 produces the letter \rightarrow ‘K’

Thereby, receiver receives the message “MOCK” that was originally sent by the sender.

6. Conclusion

In the proposed method each of the characters in the message are encrypted into an Euler graph and then secured by an Hamiltonian circuit. So, decryption is almost impossible unless the Hamiltonian circuit and the encoding scheme is known. Therefore this algorithm is secure for transmission of various messages.

Now since each graph can carry a single character, the complexity of the interpretation increases which makes it more difficult for any hacker to decode the message. Hence this proposed encryption technique is both safe and an efficient one to transfer messages.

7. References


