JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A STUDY ON GRAPH THEORY IN CRYPTOGRAPHY USING PYTHON

Dr.M.Lalitha, S.Vasu
Assistant Professor
Department of Mathematics
Kongu Arts and Science College, Erode.

Abstract

The need of secure communication of messages is nothing new. It has been present since ages. Security in today's world is one of the important challenges. Ciphers can be converted into graphs for secret communication. The field of Graph Theory plays a vital role in various fields. Especially Graph theory is widely used as a tool of encryption, due to its various properties and its easy representation in computers as a matrix. It's based on graph theory applications have been studied and we explore the usage of Graph theory in cryptography. Also we proposed python encoding and decoding to convert plain text into cipher text and cipher text into plain text.

Keywords

Euler graph, Plain text, Adjacency matrix, Cipher text, Cryptography, Encryption, Python coding.

1. Introduction

Cryptography plays a vital role in today's network. Security of data is one of the biggest concerns while exchanging data over the network. The data need to be highly confidential. Cryptography is the art of hiding data or manipulating data in such a way that where no third party can understand the original data while transmission from source to destination.

Nowadays,with the growth of network data are being exchanged widely over the networks. The basic need in every growing field is communication[1, 2]. Everyone wish to protect his or her data to be safe and secure whether it is personal or professional. The internet is mainly responsible for exchanging of data. In our day-to-day life we use many insecure way of sharing and transferring information. Cryptography is the method, which use the original information, manipulate it into another form, and share it over the network. It is used in various places such as ATM, computer passwords, military etc. When the data is transformed into another form, that form is known as 'cipher' and the changed text known as 'ciphertext'. There are various types of cryptographic methods exist which is used to encrypt or hide the secret message. [7,12]Cryptographic method uses a 'key', which is used to hide the secret message. 'Key' act as a password to lock and unlock the secret message.

Based on 'key' cryptography has many algorithms which is classified into two types: Symmetric-key algorithm and Asymmetric-key algorithm. In Symmetric-key algorithm sender and receiver shares the same key. The sender uses the key to hide or encrypt the secret message before sending it. Later, receiver use the same key to decrypt the secret message. In asymmetric-key algorithm sender and receiver shares the different key. Two types of keys used here that is 'private key' and 'public key'. The sender use the 'private-key' mostly to encrypt the secret message before sending it and receiver use the 'public-key' to decrypt the message in most of the cases. Here receiver can be many that is why public-key is being shared among all as the name suggests.

Cryptography is the science[8, 9] of secret writing with the goal of hiding the meaning of a message. Cryptography has long been the art of spies and soldiers. Nowadays, it is used every day by billions of people for securing electronic mail and payment transactions. The science of cryptography touches on many other disciplines, both within mathematics and computer science and in engineering. In mathematics, cryptology uses and touches on, algebra, number theory, graph and lattice theory, algebraic geometry and probability and statistics. Analysis of cryptographic security leads to using theoretical computer science especially complexity theory. [13, 14]The actual implementation of cryptosystems and the hard work of carrying out security analysis for specific cryptosystems falls into engineering and practical computer science and computing.

2. Preliminaries

Definition 2.1[1, 2]

A graph is formed by vertices and edges connecting the vertices. Formally, a graph is a pair of sets (V,E), where V is the set of vertices and E is the set of edges, formed by pairs of vertices.

Definition 2.2

An edge of a graph that joins a node to itself is called loop (or) Self loop.

Definition 2.3

A graph in which each edge connects two different vertices and where no two edges connect the same pair of vertices is called a simple graph.

Definition 2.4

A simple graph in which there exists an edge between every pair of vertices is called a complete graph.

Definition 2.5 [3,4]

Let V = (V, E) be a graph with

$$V = v_1, v_2, \ldots, v_n$$

$$E = e_1, e_2, \ldots, e_m$$

*a*nd without parallel edges. The adjacency matrix of G is an symmetric binary matrix $X = [x_{ij}]$ defined over the ring of integers such that

$$x_{ij} = \begin{cases} 1, if\ v_i v_j \epsilon\ E \\ 0, otherwise \end{cases}$$

It is used to represent whether a pair of vertices in a given graph are connected or not. An adjacency matrix is used to represent a finite graph. An element in the adjacency matrix is represented by x_{ij} where i represents the row and j represents the column in the adjacency matrix whose intersection is the element x_{ij}.

Definition 2.6

Let G be a graph with n vertices, m edges and without self-loops. The incidence matrix A of G is an nm matrix A = [a_{ij}] whose n rows correspond to the n vertices and the m columns correspond to m edges such that

$$a_{ij} = f(x) = \begin{cases} 1, & if\ j^{th} edge\ m_j\ is\ incident\ on\ i^{th}\ vertex \\ 0, otherwise \end{cases}$$

Definition 2.7

An Euler path is a path that travels through all edges of a connected graph.

Definition 2.8

An Euler circuit is a circuit that visits all edges of a connected graph.

Definition 2.9

A Graph is Eulerian if and only if it contains an Euler Tour. G is called an Eulerian because it contains Euler Tour. G is not Eulerian because it does not contain Euler's Tour.

Definition 2.10

A Hamiltonian path or traceable path is a path that visits each vertex of the graph exactly once. A graph that contains a Hamiltonian path is called a traceable graph. A graph is Hamiltonian connected if for every pair of vertices there is a Hamiltonian path between the two vertices.

Definition 2.11

Cryptography is the scientific and practical activity associated with developing of cryptographic security facilities of information and also with argumentation of their cryptographic resistance.

Definition 2.12

It is the message or information that the sender wishes to send to the receiver.

Definition 2.13

It is the encrypted or encoded message that contains the plain text in an unreadable format.

Definition 2.14[5]

The process of converting information or data into a code, especially to prevent unauthorized access.

Definition 2.15

Decryption is the process of receiving the plaintext from ciphertext without knowing the key.

Definition 2.16[10, 11]

In cryptography, the simple XOR cipher is a type of additive cipher, an encryption algorithm that operates according to the following principles:

$A \oplus 0 = A$

$A \oplus A = 0$

$(A \oplus B) \oplus C = A \oplus (B \oplus C)$

$(B \oplus A) \oplus A = B \oplus 0 = B$

where $\oplus$ denotes the exclusive disjunction (XOR) operation. This operation is sometimes called modulus 2 addition (or subtraction, which is identical). With this logic, a string of text can be encrypted by applying the bit wise XOR operator to every character using a given key. To decrypt the output, merely reapplying the XOR function with the key will remove the cipher.

3. Encoding Systems

The message that the user wants to send, will be encrypted into a Euler Graph bythe encryption technique. A Hamiltonian circuit will be traced out from theencrypted graph. This will be used as a key for decryption.

1. Convert each and every alphabet into its equivalent uppercase.
2. Take ASCII value of each uppercase alphabets.
3. Convert the ASCII values into binary format.
4. After the binary format of each ASCII is achieved, they are XORed with the binaryequivalent of 32.
5. Store the resultant XORed binary equivalent in an array M[i].
6. Count k= the number of 1's in the binary equivalent from the array M[i].
7. Form an adjacency matrix A = (a_{ij}) with the count k, where a_{ij} denotes the element in the i^{th} row and j^{th} column of the matrix. The count represents the number ofvertices of the simple graph. So, the principal diagonal elements of the matrix are0. This adjacency matrix is symmetric. so $a_{ij} = a_{ji}$.
8. Now count the number of 0's following the 1 for every element in the array and put$a_{ij} = a_{ji}$= number of 0's + 1. For example, if 10 is in the array the aji = 2, if 1000is in the array then $a_{ji} = 4$.
9. The above process is repeated till the Hamiltonian circuit tracing reaches the endvertex.
10. Upon reaching the last element of the array as 1 that is the binary stream ends witha 1, then make $a_{1j} = 1$ where j = k.
11. If the binary stream ends with a 1 and is followed by L number of 0's , then put $a_{1k} = L + 1$.
12. The adjacency matrix is sent to the receiver.

Alphabet	ASCII Code	Binary Number	XORed	Matrix
A	65	1000001	1100001	[1 5 1]
B	66	1000010	1100010	[1 4 2]
C	67	1000011	1100011	[1 4 1 1]
D	68	1000100	1100100	[1 3 3]
E	69	1000101	1100101	[1 3 2 1]
F	70	1000110	1100110	[1 3 1 2]
G	71	1000111	1100111	[1 3 1 1 1]
H	72	1001000	1101000	[1 2 4]
I	73	1001001	1101001	[1 2 3 1]
J	74	1001010	1101010	[1 2 2 2]
K	75	1001011	1101011	[1 2 2 1 1]
L	76	1001100	1101100	[1 2 1 3]
M	77	1001101	1101101	[1 2 1 2 1]
N	78	1001110	1101110	[1 2 1 1 2]
O	79	1001111	1101111	[1 1 2 1 1 1]
P	80	1010000	1110000	[1 1 5]
Q	81	1010001	1110001	[1 1 4 1]
R	82	1010010	1110010	[1 1 3 2]
S	83	1010011	1110011	[1 1 3 1 1]
T	84	1010100	1110100	[1 1 2 3]
U	85	1010101	1110101	[1 1 2 2 1]
V	86	1010110	1110110	[1 1 2 1 2]
W	87	1010111	1110111	[1 1 2 1 1 1]
X	88	1011000	1111000	[1 1 1 4]
Y	89	1011001	1111001	[1 1 1 3 1]
Z	90	1011010	1111010	[1 1 1 2 2]

4. Decoding Technique

1. The adjacency matrix is received.
2. The elements of the adjacency matrix are stored in a temporary array Z[p].
3. We will traverse and take into consideration either the upper triangular matrix orthelower triangular matrix along the main diagonal. This is because of the symmetricnature of the adjacency matrix.
4. To build the binary stream, the elements of the Z[p] are expanded.
5. To get back the original message, [6]the operations used in the encoding system areapplied backwards.

5. Proposed Encoding and Decoding Method

Let our plain text be GRAPH. Convert each alphabet in the plain text into binary strings.

```
import math
def toBinary(a):
        l,m = [ ],[ ]
        for i in a:
                l.append(ord(i))
        for i in l:
                m.append(int(bin(i)[2:]))
        returm m
print(""Graph" in binary is")
print(toBinary(" Graph")
```

Then XOR each alphabet with 32-bit binary string. We get the following:

```
string1 = input("enter a string:")
string2 = "0100000"
result = [(ord(a)^ord(b)) for a,b in zip(string1,string2)]
print(result)
```

The output of the above python coding is shown below:

Alphabet	ASCII Code	Binary Number	XORed
G	71	1000111	1100111
R	82	1010010	1110010
A	65	1000001	1100001
P	80	1010000	1110000
H	72	1001000	1101000

Using encryption algorithm and Euler graph, the adjacency matrix of each alphabet is constructed and shown below:

For the alphabet G, there are five 1's, therefore the graph for the alphabet G has 5 vertices and hence python coding for adjacency matrix is of 5x5 order given below:

```
# Add a vertex to the set of vertices and the graph
def add vertex(v):
        global graph
        global vertices no
        global vertices
        if v in vertices:
                print("Vertex", v,"already exists")
```

```
        else:
                vertices_no = vertices_no + 1
                vertices.append(v)
                if vertices_no>1:
                        for vertex in graph:
                                vertex.append(0)
                temp = [ ]
                for i in range(vertices_no):
                        temp.append(0)
                graph.append(temp)
# Add an edge between vertex v1 and v2 with edge weight e
def add edge(v1, v2, e):
        global graph
        global vertices_no
        global vertices
        # Check if vertex v1 is a valid vertex
        if v1 not in vertices:
                print("Vertex", v1," does not exist.")
        # Check if vertex v1 is a valid vertex
        elif v2 not in vertices:
                print("Vertex", v2," does not exist.")
        # Since this code is not restricted to a directed or
        # An undirected graph, an edge between v1 v2 does not
        # Imply that an edge exists between v2 and v1
        else:
                index1 = vertices.index(v1)
                index2 = vertices.index(v2)
                graph[index1][index2] = e
# Print the graph
def print graph():
        global graph
        global vertices_no
        for i in range(vertices_no):
                for j in range(vertices no):
                        if graph[i][j] != 0:
                                print(vertices[i], " ->", vertices[j], \
                                " edge weight: ", graph[i][j])
```

```
# Driver code # stores the vertices in the graph
vertices = [ ]
# stores the number of vertices in the graph
Vertices_no = 0
graph = [ ]

# Add vertices to the graph
add vertex(1)
add vertex(2)
add vertex(3)
add vertex(4)
add vertex(5)
# Add the edges between the vertices by specifying
# the from and to vertex along with the edge weights.
add edge(1, 2, 1)
add edge(1, 5, 1)
add edge(2, 1, 1)
add edge(2, 3, 3)
add edge(3, 2, 3)
add edge(3, 4, 1)
add edge(4, 3, 1)
add edge(4, 5, 1)
add edge(5, 1, 1)
add edge(5, 4, 1)
print graph()
print("Internal representation: ", graph)
```

Therefore, its adjacency matrix is

$$G = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Similarly for the other alphabets, we get

$$R = \begin{pmatrix} 0 & 1 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 3 \\ 2 & 0 & 3 & 0 \end{pmatrix}, A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 5 \\ 1 & 5 & 0 \end{pmatrix}, P = \begin{pmatrix} 0 & 1 & 5 \\ 1 & 0 & 1 \\ 5 & 1 & 0 \end{pmatrix}, H = \begin{pmatrix} 0 & 1 & 4 \\ 1 & 0 & 2 \\ 4 & 2 & 0 \end{pmatrix}$$

All these matrices are sent to the receiver one by one.

For the first matrix the receiver receives [1 3 1 1 1]

For the second matrix the receiver receives [1 1 3 2]

For the 3rd matrix the he receives [1 5 1]

For the 4th matrix the he receives [1 1 5]

For the 5th matrix the he receives [1 2 4].

By expanding these matrices, the receiver gets

13111 = 1100111

1132 = 1110010

151 = 1100001

115 = 1110000

124 = 1101000

The python coding for decoding technique is,

```
import math
def tostring(a):
        l = [ ]
        m = ""
        for i in a:
                b = 0
                c = 0
                k = int(math.log10(i))+1
                for j in range(k):
                        b=((i%10)*(2**j))
                        i=i/10
                        c=c+b
                l.append(c)
        for x in l:
                m=m+chr(x)
        return m
print("\n"[1100111, 1110010, 1100001, 1110000, 1101000]" in stringis ")
print(tostring([1100111, 1110010, 1100001, 1110000, 1101000]))
```

Using this decoding technique is

1100111 gives G

1110010 gives R

1100001 gives A

1110000 gives P

1101000 gives H.

Hence the receiver receives the plain text which was originally sent by the sender.

CONCLUSION

The cryptography is an algorithm which provides secure communication. In this paper, we studied a technique where each character of the data will be encrypted into an Euler Graph. Hamiltonian Circuit is used as key to secure the data. In this cryptography technique, the complexity and the uncertainty of the decryption and interpretation of the actual message is very high and difficult as each graph represents a character of the message. Here converted plain text into cipher text and cipher text into plain text by using the proposed python coding technique. This technique developed the algorithm. This algorithm ensures the safety of the data.

REFERENCES

[1]AleksanderMaricq, Applications of Expander Graphs in Cryptography, May 16, 2014.

[2] P. Amudha, A.C. Charles Sagayaraj, A.C. Shantha Sheela, An Application of Graph Theory in Cryptography, 2015.

[3] Amrutha, R. Thandeeswaran, N. Jeyanthi, Cloud based VoIPApplication in Aircraft Data Networks, International Journal of GridDistribution Computing, Vol.7, No.6 (2014) December, pp.11- 18,2014.

[4] F.Amounas. ”Enhanced Elliptic Curve Encryption Approach ofAmazigh alphabet with Braille representation”, International Journal ofComputer Science and Network Solutions, vol 3, No 8, pp. 1-9. 2015.

[5] N.Jeyanthi, R. Thandeeswaran, J. Vinithra, RQA BasedApproach to Detect and Prevent DDoS Attacks in VoIP Networks,Cybernetics and Information Technologies, Vol.14, No.1, pp. 11-24, 2014.

[6] N.Koblitz, Algebraic Aspects of Cryptography, Springer-Verlag, Berlin 1998.

[7] MahanteshGawannavar, Payal Mandulkar, R. Thandeeswaran, N. Jeyanthi, Office in cloud: Approach to Authentication and Authorization, Recent Advances in Communications and Networking Technology, Bentham sciences, Vol.4, No.1, pp.49-55, 2015.

[8] Manisha Kumari, Data Encryption and Decryption Using Graph Plotting, Department of Computer Science, Christ University, Bangalore, India, 2010.

[9] Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice Hall, 2010.

[10] NataliaTokareva, Connections between graph theory and cryptography, G2C2: Graphs and Groups, Cycles and Coverings Novosibirsk, Russia,September, 24–26, 2014.

[11] Rishi Pal Singh, Vandana , Application of Graph Theory in ComputerScience and Engineerin International Journal of Computer Applications(0975 8887) Volume 104 No.1, October 2014.

[12] Sarah Simpson, Cryptography in Everyday Life, This document submitted as partial requirement for ECO350k, Spring 1997.

[13] SivaKishore.B, Siva ThejaReddy.J, N.Jeyanthi, Three Phase Power Management Algorithms for Green Cloud Computing, International Journal of Applied Engineering Research Vol. 8, No.14, pp. 1725-1736,2013.

[14] Wael Mahmoud Al Etaiwi, Encryption Algorithm Using Graph Theory, Information Technology Directorate, Jordan Customs Department, Amman, Jordan, 2012.