© 2016 IJSRSET | Volume 2 | Issue 3 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

An Innovative Approach for Enhancing the Security of Amazigh Text Using Graph Theory Based ECC

Fatima AMOUNAS
R.O.I Group, Computer Sciences Department, Moulay Ismail University, Faculty of Sciences and Technics, Errachidia, Morocco

ABSTRACT

Security in today's world is one of the important challenges. Encryption is one of the popular methods to achieve secret communication between sender and receiver. Graph theory is widely used as a tool of encryption, due to its various properties and its easy representation in computers as a matrix. In this paper we introduce an enhanced approach of elliptic curve encryption algorithm for achieving better data protection using graph theory. Experimental results show that this proposed method is more efficient and robust.

Keywords: Elliptic curve, Cryptography, Graph theory, Adjacency Matrix, Unicode, Amazigh text.

I. INTRODUCTION

With the development of computers, researchers have gathered special interest in the cryptographic algorithms to protect information. Cryptography is one of the mathematical techniques that ensure secure communications within a non-secure channel. Cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break by any adversary. Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields [1]. ECC can be defined over two types of fields: one is the prime field Fp which is suitable for the software applications and the other is the binary field which is suitable for the hardware applications. ECC offers the same security level like RSA and ElGamal algorithms with shorter key length which makes it works with a little amount of memory and low power. For this reason, most of the modern cryptographic systems are established based on the elliptic curve. For instance, Ali Makki in 2012 proposed three techniques based on the elliptic curve [2]. The author used multiplication operation instead of inverse operation in order to reduce the calculation time compared with the original method. Meltem Kurt and Tarik Yerlikaya in 2013 presented a modified cryptosystem using hexadecimal to encrypt data [3].

Recently, Graph theory has a great contribution in the development of various encryption techniques. Several methods have been presented by many researchers. M. Yamuna and al. in 2013 used a musical notes along with graph theory to encrypt binary messages [4]. M. Yamuna and al. in 2014 proposed a method of encrypting any four digit pin number as a digraph [5]. This method is efficient and can be used for multiple pin number communication. Shubham Agarwal and Anand Singh Uniyal in 2015 proposed an efficient encryption scheme using prime weighted graph in cryptographic system for secure communication [6]. Krishnappa.H.K. and al. [7] showed that how to use magic squares to realize vertex magic total labeling of complete graph.

Al Etaiwi W.M. [8] presented a new cryptography algorithm based on graph theory. The author provides an encryption algorithm using an encoding table based on ASCII value and graph properties. In this context, the proposed paper attempts to enhance the efficiency by providing add on security to the elliptical cryptosystem using graph properties. More precisely, this paper proposes the practical application of graph theory in the field of cryptography.

This paper is arranged as follows. Section 1 is an introduction. Section 2 investigates the basic theory of elliptic curve, graph theory and provides an overview of Amazigh alphabet. Section 3, presents the proposed approach. Section 4 is devoted to the illustration with an example, followed by experimental results in section 5. Finally, section 6 concludes the paper.

II. METHODS AND MATERIAL

1. Background Information

A. Elliptic Curve

Elliptic Curve Cryptography was introduced in 1986 by Victor Miller and Neil Kolbitz as an alternative to other public key cryptosystem present such as RSA. The mathematical background of ECC is more complex and thus it provides greater security and more efficient performance than other public key cryptosystems [9].

1) Definition: An elliptic curve over a field K is the set of points satisfying the Weierstrass equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad (1)$$

and also an element denoted Ω and called the point at infinity, where a_1, a_3, a_2, a_4, a_6 $\in$ K.

For fields of various characteristics, the Weierstrass equation Eq.(1) can be transformed into different forms by a linear change of variables.

For the homogeneous, so called also projective, coordinate system, $(x, y) = (X/Z/, Y/Z)$, the equation (1) determining an elliptic curve point takes the following form:

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3 \qquad (2)$$

An elliptic curve E over a finite field K can be made into an abelian group by defining an additive operation on its points [10].

2) Rules for Addition: suppose P(x_1, y_1) and Q(x_2, y_2) are two points lie on an elliptic curve E defined in Equation 1.

The sum P+Q results a third point R(x_3, y_3) which is also lies on E. Then R is given by:

- R = Ω pour $x_1 = x_2$ and $y_2 = -y_1 - a_1x_1 - a_3$.
- $x_3 = t^2 + a_1t - a_2 - x_1 - x_2$ and $y_3 = -(t + a_1)x_3 - s - a_3$

where

$$t = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, \text{ if } P \neq Q. \\[2ex] \dfrac{3x_1^2 + 2a_1x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}, \text{ if } P = Q. \end{cases}$$

and

$$s = \begin{cases} \dfrac{y_1x_2 - y_2x_1}{x_2 - x_1}, \text{ if } P \neq Q. \\[2ex] \dfrac{-x_1^3 + a_4x_1 + 2a_6 - a_3y_1}{2y_1 + a_1x_1 + a_3}, \text{ if } P = Q. \end{cases}$$

The addition operation defined above turns E(K) into an abelian group that has Ω as the identity element.

3) Scalar multiplication: Let k be an integer and P is a point lies on E. The scalar multiplication can be defined by

kP = P + P + ::: + P(k times).

B. Graph Theory Basics

Graph theory is a branch of applied mathematics. Fundamentally, a graph consists of a set of vertices, and a set of edges, where an edge is something that connects two vertices in the graph. A graph is a pair (V, E), where V is a finite set and E is a binary relation on V. V is called a vertex set whose elements are called vertices. E is a collection of edges, where an edge is a pair (u,v) with u,v in V. Graph G =(V, E) is a collection of V nodes connected by E links.

Path: A path is a simple graph whose vertices can be ordered so that two vertices are adjacent if and only if they are consecutive in the list.

Undirected Graph: A graph in which each edge symbolizes an unordered, transitive relationship between two nodes. Such edges are rendered as plain lines or arcs.

Directed Graph: A graph in which each edge symbolizes an ordered, non-transitive relationship between two nodes. Such edges are rendered with an arrowhead at one end of a line or arc.

Weighted edge: Weighted edges symbolize relationships between nodes which are considered to have some value, for instance, distance or lag time. Such edges are usually annotated by a number or letter placed beside the edge. If edges have weights, we can put the weights in the lists.

Tree: An undirected connected graph T is called tree if there are no cycles in it. There is exactly one simple path between any vertices u and v.

Simple path: Simple path is a path in which all the vertices are distinct.

Spanning Tree: A spanning tree T of a graph G is a sub graph containing all the nodes of G. It is a minimal set of edges that connects all the nodes of G without creating any cycles or loops. Out of all the spanning trees of G, the minimum spanning tree is one with least weight.

Shortest Path Algorithms:

- *Dijkstras' Algorithm:*

- Finds single-source shortest path in weighted graph.
- It replaces the Breadth First Search (BFS) queue with a Priority Queue. Vertices are added to the Priority Queue by their distance away from the source [11].
- If negative weight is used, Dijkstra's algorithm might fail. Dijkstra's algorithm does not work with negative weight arcs.

- *Floyd-Warshall's Algorithm:*

- Finds all-pair shortest path in weighted graph.
- Uses Adjacency matrix.
- This algorithm compares all possible paths through the graph between each pair of vertices.
- Negative weights are allowed but Negative cycle is not allowed.

- *Bellman-Ford Algorithm:*

- Finds single-source shortest path in weighted graph and detects negative cycles.
- Its basic structure is very similar to Dijkstra's algorithm, but instead of greedily selecting the minimum-weight node not yet processed to relax, it simply relaxes all the edges. The repetitions allow minimum distances to accurately propagate throughout the graph, since, in the absence of negative cycles; the shortest path can only visit each node at most once.

C. Amazigh Language

During the last few decades, most researches have focused on standardization of Amazigh language. This process was initiated by character encoding specified by extended ASCII, incorporation into Unicode standard [12] and implementation of a standard keyboard layout. A phonetic mapping was set between Latin, Arabic and Tifinagh [13], which has facilitated the passage from a graphic system to another, of most concern here between Tifinagh and Latin where directionality is the same (heading towards the right). Since 2004, the recommended encoding for Tifinagh is the one set up by ISO 10646/UNICODE. Hence, Tifinagh is part of the Unicode range, from U+2D30 to U+2D7F.
Thereby, Amazigh language has become an official language in Morocco in addition to Arabic.
The Amazigh language has its own writing system called Tifinaghe-IRCAM [14]. This alphabet includes:

- 27 consonants:
  - The labials:ⴼ, ⴱ, ⵎ.
  - The dental: ⵜ, ⴷ, ⴹ, ⵟ, ⵏ, ⵔ, ⵕ, ⵍ.
  - The alveolar:ⵙ, ⵣ, ⵚ, ⵥ.
  - The palatal: ⵛ, ⵊ.
  - the Velar: ⴽ, ⴳ.
  - the labiovelars: ⴽⵯ, ⴳⵯ.
  - the uvular: ⵇ, ⵅ, ⵖ.
  - the pharyngeal: ⵄ, ⵃ.
  - the laryngeal: ⵀ.
- 2 semi consonants: ⵢ, ⵡ.
- 4 vowels: ⴰ, ⵉ, ⵓ, ⴻ.

2. Our Contribution

The proposed system applies some manipulations to the original data using graph theory and some of its properties. The main idea of our contribution depends on using graph theory to generate the complete weighted graph. More precisely, this paper presents a novel way to label the edges of the graph. Then, apply ECC operations based matrix approach to generate the strong cipher text.

The complete graph G is represented as an (n×n) matrix, in which all the entries of the principle diagonal are used to label the vertices of the graph.

Let us label the edges as points on elliptic curve. The construction of the labeling matrix for complete graph with n vertices is given as follows:

$$PM = \begin{pmatrix} P_1 & P_{1,2} & \dots & P_{1,n} \\ P_{2,1} & P_2 & \dots & P_{2,n} \\ & & . & \\ & & . & \\ & & . & \\ P_{n,1} & P_{n,2} & \dots & P_n \end{pmatrix}$$

where $P_{i,j} = P_j - P_i$, with $P_j, P_i \in EC$.

Here, the entries of data matrix are points on elliptic curve.

A. Encryption/Decryption

Assume that both of the sender (user A) and the receiver (user B) agreed on the use of the elliptic curve given by:

$$E : y^2 = x^3 + ax + b \mod p \qquad (3)$$

Every entity needs to choose a private key. The private keys are denoted n_A and n_B respectively. The public keys can be generated as follows: $P_A = n_A P \quad P_B = n_B P$.
The secret key will be $K = n_A P_B = n_B P_A$.
The proposed method requires both the sender of the message and the receiver of the message to know the following relationships:

$E(F_p)$: the set of points on elliptic curve.
P: base point with order N.
S1: the set of all alphabets.
S2: the set of the mapping points.

We define the mapping F: S1 →S2, as specified rule of correspondence between sets of characters which are composed original message and a set of points on elliptic curve [15].

1) Encryption process

Step 1. Imbed each character into point on elliptic curve.

Step 2. Add vertex for each character in the plain text to the graph G.

Step 3. Link the vertices together by adding an edge between each sequential character in the original message until we form a cycle graph.

Step 4. Draw edges between the vertex pairs (v_i, v_j). Let us label these edges as P_{ij} obtained as follows:

$$P_{ij} = P_j - P_i.$$

Step 5. Add more edges to form a complete graph G. Then, assign d_{ij} as the edge weight to the edge so that $P_{ij}=d_{ij}P$ (by solving the discrete logarithm) and generate its adjacency matrix M.

Step 6. Compute a secure key K and modify the graph by assigning the new weighted values: $w_i=d_{ij} + x_K$ such that x_k is the x-coordinate of K.

Step 7. Find the minimum spanning tree and create its adjacency matrix M_1.

Step 8. Construct the labelling matrix PM for the complete graph.

Step 9. Modify a matrix PM by adding a secure key to the entries of diagonal.

$$PM' = (p_{ij}) = \begin{cases} P_{ii} + K, & i = j. \\ P_j - P_i, & i \neq j. \end{cases}$$

Step 10. Multiply matrix M_1 by PM' to get C.

Step 11. Send the result matrix to the receiver in a linear format (ie. either column wise or row wise) with space between each between element.

Cipher= (cl, n, M', C)

where
cl: 1 or 2 (1 represents row, 2 represents column).
n: size of matrix.
M': upper triangular part of the adjacency matrix of the complete graph G.

2) Decryption process

Step 1. Read the encrypted data and form the required matrices.

Step 2. Construct the adjacency matrix from M' and hence the corresponding complete graph from the matrix.

Step 3. Compute a secure key K and assign the new weighted values: $w_i=d_{ij} - x_K$ to the edge.

Step 4. Find the minimum spanning tree and its adjacency matrix M_1.

Step 5. Multiply matrix C by M_1^{-1} to get PM'

Step 6. Construct a data matrix PM from PM' such that:

$$p_{ii} = p'_{ii} - K.$$

Step 7. Extract the points on the diagonal and reverse the embedding to get the original message.

III. RESULTS AND DISCUSSION

1. Implementation Example

Assume that the communicating parts are agreed to use the elliptic curve:

$$E : y^2 = x^3 + x + 13 \bmod 53 \qquad (4)$$

The set of all points on elliptic curve is shown below in Figure 1.

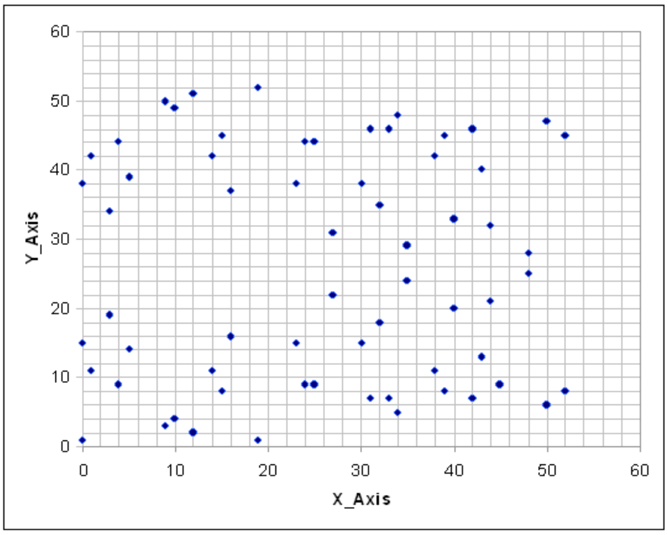


Figure 1. The set of points on elliptic curve $E(F_{53})$.

Let the point (4, 9) be chosen as the base point.
If user A wants to send the message “ϯₒX8OƸ” to user B, he does the following:

First, convert the message to a graph, by converting each character to a vertex. Then, link each two sequential characters together to form a cycle graph as shown in Figure 2.

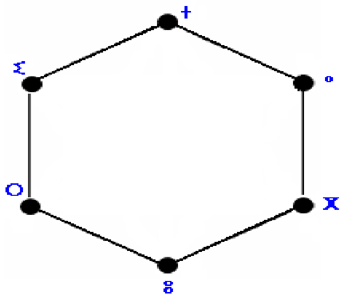


Figure 2. Graph contains the characters of the original message.

After that constructs the complete graph and weights each edge using represents the difference between the corresponding points of the connected two vertices.

Now, add more edges to obtain the complete weighted graph as shown in Figure 3.

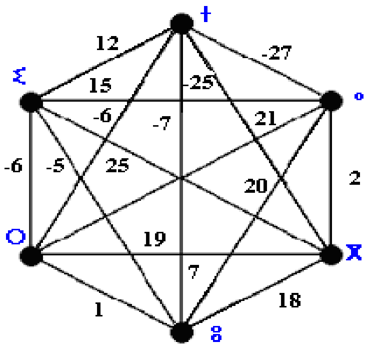


Figure 3. Complete weighted graph.

The adjacency matrix of the complete graph is given as follows:

$$M=\begin{pmatrix} 0 & -27 & -25 & -7 & -6 & -12 \\ 27 & 0 & 2 & 20 & 21 & 15 \\ 25 & -2 & 0 & 18 & 19 & 13 \\ 7 & -20 & -18 & 0 & 1 & -5 \\ 6 & -21 & -19 & -1 & 0 & -6 \\ 12 & -15 & -13 & 5 & 6 & 0 \end{pmatrix}$$

Let K= (12,2) be the chosen secure key. Now the modified graph is shown in Figure 4.

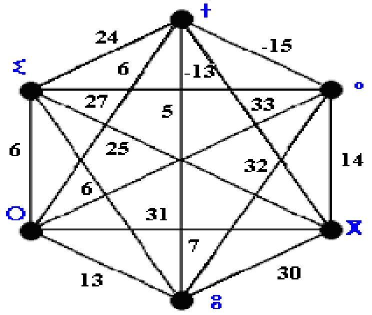


Figure 4. Complete weighted graph.

The above weighted graph is converted into the following spanning tree:

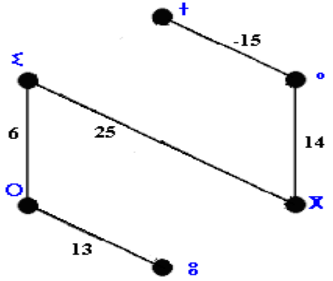


Figure 5. Minimum spanning tree graph.

The result adjacency matrix is given as follows:

$$M_1=\begin{pmatrix} 0 & -15 & 0 & 0 & 0 & 0 \\ -15 & 0 & 14 & 0 & 0 & 0 \\ 0 & 14 & 0 & 25 & 0 & 0 \\ 0 & 0 & 25 & 0 & 6 & 0 \\ 0 & 0 & 0 & 6 & 0 & 13 \\ 0 & 0 & 0 & 0 & 13 & 0 \end{pmatrix}$$

Next, construct a labeling data matrix with entries are points on elliptic curve:

$$PM=\begin{pmatrix} (15, 8) & (24,44) & (0, 38) & (5, 39) & (43,40) & (1, 42) \\ (25,44) & (4, 9) & (30,38) & (40,33) & (33, 7) & (52,45) \\ (38,42) & (4, 44) & (44,32) & (35,29) & (3, 19) & (39,45) \\ (16,37) & (3, 34) & (35,24) & (33, 7) & (4, 9) & (42, 7) \\ (5, 14) & (40,20) & (35,24) & (45, 9) & (23,38) & (43,40) \\ (19, 1) & (14,42) & (19,52) & (43,13) & (5,14) & (34,38) \end{pmatrix}$$

After that compute a secure key K= (12, 2) and modify PM to obtain PM'.

Now, multiply the data matrix PM' by M_1 to form C.

$$C=\begin{pmatrix} (40,20) & (30,38) & (27,31) & (40,33) & (43,13) & (48,25) \\ (1,11) & (5, 39) & (34, 5) & (38,42) & (0,15) & (33,46) \\ (23,15) & (32,35) & (4, 9) & (12,51) & (45, 9) & (33, 7) \\ (33, 7) & (34,48) & (35,29) & (44,21) & (24,44) & (27,31) \\ (5, 14) & (43,13) & (24,44) & (43,13) & (40,33) & (48,25) \\ (14,11) & (34, 5) & (50, 6) & (19,52) & (1, 11) & (39, 8) \end{pmatrix}$$

Therefore, the encrypted data is:

2 6 -27 -25 2 -7 20 18 -6 21 19 1 -12 15 13 -5 -6 40 20 1 11 23 15 33 7 5 14 12 11 30 38 5 39 32 35 34 48 43 13 34 5 27 31 34 5 4 9 35 29 24 44 50 6 40 33 38 42 12 51 44 21 43 13 19 52 43 13 0 15 45 9 24 44 40 33 1 11 48 25 33 46 33 7 27 31 48 25 39 8

After receiving the cipher text, the receiver extracts the first entry that represents the data values are to be read column wise. The second entry represents the graph has 6 vertices and the size of the adjacency matrix is 6×6. The adjacency matrix is constructed from the receiving upper matrix as follows:

$$M=\begin{pmatrix} 0 & -27 & -25 & -7 & -6 & -12 \\ 27 & 0 & 2 & 20 & 21 & 15 \\ 25 & -2 & 0 & 18 & 19 & 13 \\ 7 & -20 & -18 & 0 & 1 & -5 \\ 6 & -21 & -19 & -1 & 0 & -6 \\ 12 & -15 & -13 & 5 & 6 & 0 \end{pmatrix}$$

The remaining data represents a data matrix C.
After that, construct the complete weighted graph G.
Next, compute secure key K= (12, 2) and modify the complete graph with the new values ($w_i=d_{ij}-x_K$).

Now, find the minimum spanning tree and its adjacency matrix M_1.

After that compute $M_1^{-1}C$ to retrieve PM'= (p'_{ij}).

$$PM'=\begin{pmatrix} (30,15) & (24,44) & (0, 38) & (5, 39) & (43,40) & (1, 42) \\ (25,44) & (27,31) & (30,38) & (40,33) & (33, 7) & (52,45) \\ (38,42) & (4, 44) & (25, 9) & (35,29) & (3, 19) & (39,45) \\ (16,37) & (3, 34) & (35,24) & (50,47) & (4, 9) & (42, 7) \\ (5, 14) & (40,20) & (35,24) & (45, 9) & (9, 3) & (43,40) \\ (19, 1) & (14,42) & (19,52) & (43,13) & (5,14) & (14,42) \end{pmatrix}$$

Then, and Modify PM' to retrieve PM: $P_{ii}=P'_{ii}-K$.

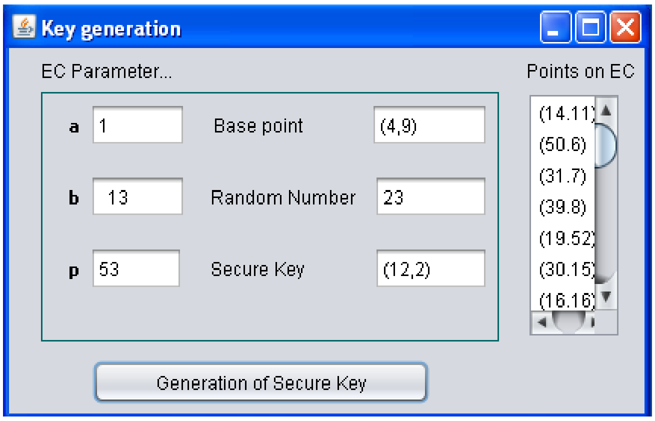
The resulting points from the diagonal are: (15, 8) (4, 9) (44, 32) (33, 7) (23, 38) (34, 38)

Now, reverse the embedding to retrieve the original message "ƚₒӾ8OƩ".

2. Result

This section demonstrates the proposed algorithm in practical aspect using Java programming language [16]. The proposed algorithm is implemented as a project for encrypting and decrypting both file and data. A Java Swing application is developed using Netbeans 7.1 to implement this methodology. The following set of figures shows the encryption and decryption of text message and text file.

The main interface of the application is illustrated in Figure 6. The user should enter the elliptic curve parameters (a, b, p), selects the base point on EC and chooses a random number. By clicking on "Generate Secure Key" button, the key generation process is performed.



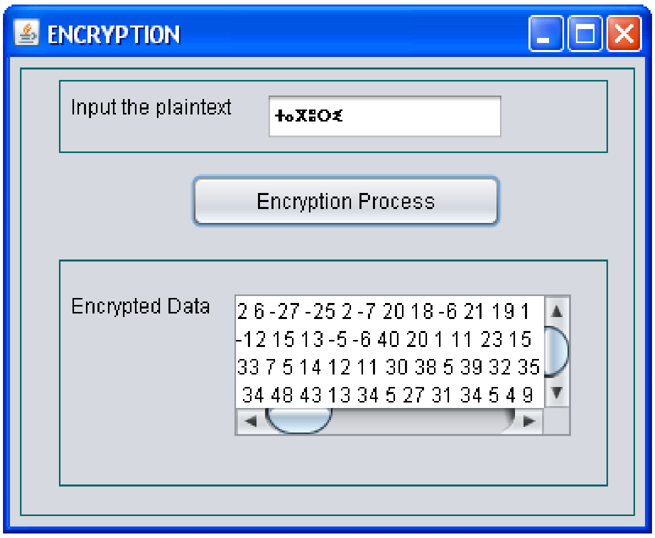
a: 1
Base point: (4,9)
b: 13
Random Number: 23
p: 53
Secure Key: (12,2)

Points on EC: (14,11) (50,6) (31,7) (39,8) (19,52) (30,15) (16,16)

Figure 6. Generation of secure key.

A. Encryption/Decryption of text message

To encrypt the text message input any Amazigh sentence in the Text Box and then click "encryption Process" button to perform encryption and the result is displayed in the resultant text box.



Input the plaintext: ⵜⴰⵅⵓⵔⵉ

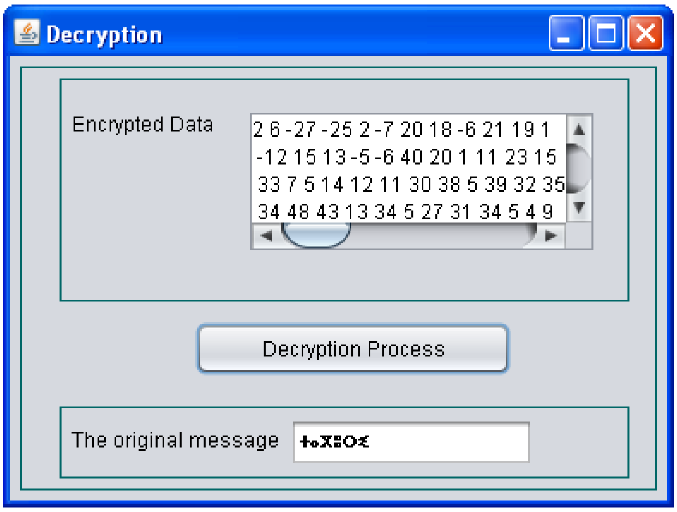
Encrypted Data:

```
2 6 -27 -25 2 -7 20 18 -6 21 19 1
-12 15 13 -5 -6 40 20 1 11 23 15
33 7 5 14 12 11 30 38 5 39 32 35
34 48 43 13 34 5 27 31 34 5 4 9
```

Figure 7. Encryption Process

The decryption is the reverse process of encryption, in which the cipher text is converted to plain text.

To decrypt the encrypted data, just click in "Decryption process" button and the cipher text is converted to original message (plaintext).



Encrypted Data:

```
2 6 -27 -25 2 -7 20 18 -6 21 19 1
-12 15 13 -5 -6 40 20 1 11 23 15
33 7 5 14 12 11 30 38 5 39 32 35
34 48 43 13 34 5 27 31 34 5 4 9
```

The original message: ⵜⴰⵅⵓⵔⵉ

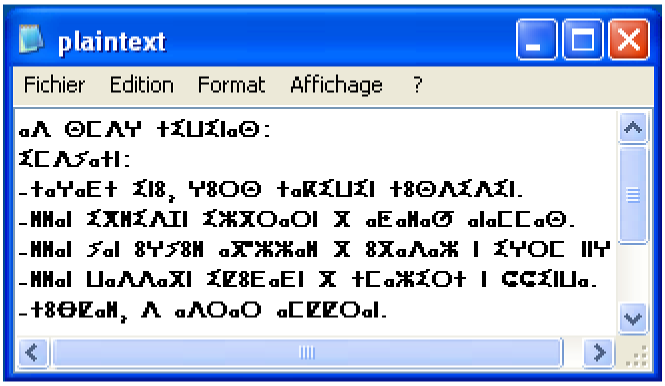
Figure 8. Decryption Process.

B. Encryption/Decryption of text file

The proposed method is also implemented for encrypting the files. Figure 9 shows the original input file to be encrypted. Here, the text file is a collection of Amazigh sentences from the schoolbook.

Files used in the program are shown below:

- Plaintext.txt: contains the text to be encrypted.
- Ciphertext.txt: contains the encrypted data.
- Original.txt: contains the decrypted data.

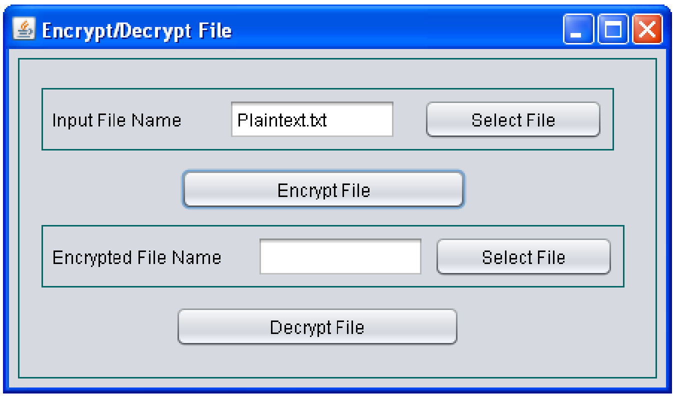


ⴰⴷ ⵙⵍⴷⵖ ⵜⵉⵡⵉⵏⴰⵙ:
ⵉⵍⴷⵣⴰⵜⵏ:
-ⵜⴰⵖⴰⵎⵜ ⵉⵏⴱ, ⵖⴱⵔⵙ ⵜⴰⴽⵉⵡⵉⵏ ⵜⴱⵙⴷⵉⴷⵉⵏ.
-ⵀⵀⴰⵏ ⵉⵅⵀⵉⴷⵊⵏ ⵉⵥⵅⵔⴰⵔⵏ ⵅ ⴰⵎⴰⵀⴰⵚ ⴰⵏⴰⵍⵍⴰⵙ.
-ⵀⵀⴰⵏ ⵣⴰⵏ ⴱⵖⵣⴱⵀ ⴰⵅⵥⵥⴰⵀ ⵅ ⴱⵅⴰⴷⴰⵥ ⵏ ⵉⵖⵔⵍ ⵏⵖ
-ⵀⵀⴰⵏ ⵡⴰⴷⴷⴰⵅⵏ ⵉⵒⴱⵎⴰⵎⵏ ⵅ ⵜⵍⴰⵥⵉⵔⵜ ⵏ ⵛⵛⵉⵏⵡⴰ.
-ⵜⴱⵚⵒⴰⵀ, ⴷ ⴰⴷⵔⴰⵔ ⴰⵍⵒⵒⵔⴰⵏ.

Figure 9. Text File to encrypt.

To encrypt the text file, the user browses the file to be encrypted and fetch the original file as shown in Figure 10. Here, the text file is divided into small blocks of messages. Each block is encrypted separately using the proposed approach. Then, we concatenate cipher texts to form the whole message cipher text.

The procedure to perform this is, the file is encrypted on clicking the "encrypt File" button and a new encrypted-file is generated as shown in Figure 11.

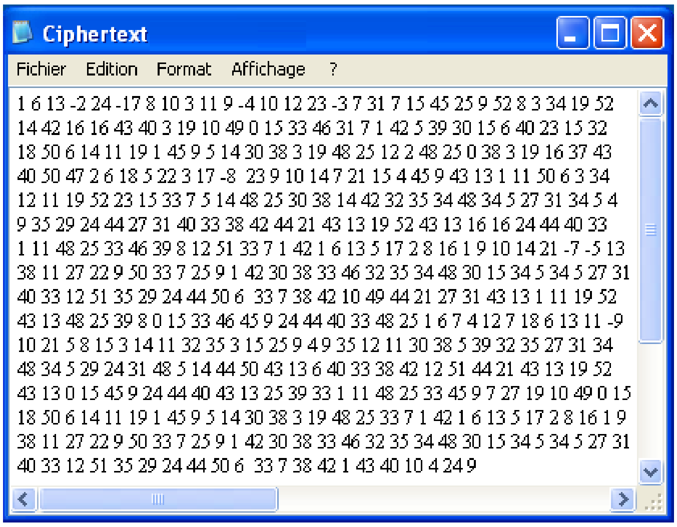


Input File Name: Plaintext.txt

Encrypted File Name:

Figure 10. Encryption and Decryption process of Text File.

The result encrypted file is shown as follows:



```
1 6 13 -2 24 -17 8 10 3 11 9 -4 10 12 23 -3 7 31 7 15 45 25 9 52 8 3 34 19 52
14 42 16 16 43 40 3 19 10 49 0 15 33 46 31 7 1 42 5 39 30 15 6 40 23 15 32
18 50 6 14 11 19 1 45 9 5 14 30 38 3 19 48 25 12 2 48 25 0 38 3 19 16 37 43
40 50 47 2 6 18 5 22 3 17 -8  23 9 10 14 7 21 15 4 45 9 43 13 1 11 50 6 3 34
12 11 19 52 23 15 33 7 5 14 48 25 30 38 14 42 32 35 34 48 34 5 27 31 34 5 4
9 35 29 24 44 27 31 40 33 38 42 44 21 43 13 19 52 43 13 16 16 24 44 40 33
1 11 48 25 33 46 39 8 12 51 33 7 1 42 1 6 13 5 17 2 8 16 1 9 10 14 21 -7 -5 13
38 11 27 22 9 50 33 7 25 9 1 42 30 38 33 46 32 35 34 48 30 15 34 5 34 5 27 31
40 33 12 51 35 29 24 44 50 6  33 7 38 42 10 49 44 21 27 31 43 13 1 11 19 52
43 13 48 25 39 8 0 15 33 46 45 9 24 44 40 33 48 25 1 6 7 4 12 7 18 6 13 11 -9
10 21 5 8 15 3 14 11 32 35 3 15 25 9 4 9 35 12 11 30 38 5 39 32 35 27 31 34
48 34 5 29 24 31 48 5 14 44 50 43 13 6 40 33 38 42 12 51 44 21 43 13 19 52
43 13 0 15 45 9 24 44 40 43 13 25 39 33 1 11 48 25 33 45 9 7 27 19 10 49 0 15
18 50 6 14 11 19 1 45 9 5 14 30 38 3 19 48 25 33 7 1 42 1 6 13 5 17 2 8 16 1 9
38 11 27 22 9 50 33 7 25 9 1 42 30 38 33 46 32 35 34 48 30 15 34 5 34 5 27 31
40 33 12 51 35 29 24 44 50 6  33 7 38 42 1 43 40 10 4 24 9
```

Figure 11. Encrypted File “ciphertext.txt”

The procedure to decrypt the file is: user browses the file to be decrypted and fetch the encrypted file (Figure10). The file is decrypted on clicking the “decrypt File” button and the decrypted file is saved in Original.txt.

IV. CONCLUSION

Graph theory is growing as a promising field in various fields. It is currently developing into an important tool for encryption and decryption due to its easy representation in computers and the properties satisfied by graphs. The proposed algorithm represents a new approach to encrypt and decrypt Amazigh text securely based elliptic curve with the benefits of graph theory properties. Our proposed approach is the combination of elliptic curve cryptography and graph theory. Hence it is much more secure and much more efficient to put the unauthorized person in a difficult position in the context of accessing the secure data. Finally, we like to point out that the technique to label the edges of the graph with elliptic curve will provide better performance in this regard.

V. REFERENCES

[1] Darrel R. Hankerson, A. Menezes and A. Vanstone, “Guide to Elliptic Curve Cryptography”, Springer, 2004.

[2] Ali M. Sagheer, “Elliptic curves cryptographic techniques”, proceeding of the 6th International Conference on Signal Processing and Communication Systems, 2012 IEEE, Gold Coast, Australia, , pp.12-14 December 2012.

[3] Kurt M, Yerlikaya Y. 2013. “A new modified cryptosystem based on Menezes Vanstone elliptic curve cryptography algorithm that uses characters hexadecimal values". International Conference on Technological Advances in Electrical, Electronics and Computer Engineering: IEEE, pp. 449-453.

[4] M. Yamuna, A. Sankar, Siddarth Ravichandran, V. Harish. 2013. “Encryption of a Binary String Using Music Notes and Graph theory”, International Journal of Engineering and Technology, Vol. 5 No 3, pp. 2920-2925.

[5] M. Yamuna, A. Suwathi, Nikitha Krishnan. 2014. “Four Digit Pin Number as a Digraph”, International Journal of Computer Application, Issue 4, Volume 3, pp. 100-107.

[6] Shubham Agarwal and Anand Singh Uniyal. 2015. “Prime Weighted Graph in Cryptographic system for Secure Communication”, International Journal of Pure and Applied Mathematics, Vol. 105 No. 3, pp. 325-338.

[7] Krishnappa.H.K., N.K.Srinath and P.Ramakanth Kumar. 2010. “Vertex Magic Total Labeling of Complete graphs”, IJCMSA., Vol 4, No 1-2, pp. 157-169.

[8] Wael Mahmoud Al Etaiwi. 2014, “Encryption Algorithm Using Graph Theory”, Journal of Scientific Research & Reports, Journal of Scientific Research & Reports, Vol 3, No.19, pp. 2519-2527.

[9] Sagheer AM. “Enhancement of elliptic curve cryptography methods [MSc Thesis], Iraq, University of Technology, 2004.

[10] Hoffstein, Jeffrey, Jill Pipher, and Joseph H. Silverman. "In An Introduction to Mathematical Cryptography", pp. 299-371. Springer, 2014.

[11] Thippeswamy.K., Hanumanthappa. J., Manjaiah D.H. 2010. “A study on contrast and comparison between Bellman-Ford and Dijkstra’s Algorithms”, National Conference on wireless Networks.

[12] Ameur M. 2006. “Graphie et orthographe de l’amazighe”, Rabat: IRCAM, pp. 47-48.

[13] Zenkouar L. 2008. "Normes des technologies de l’information pour l’ancrage de l’écriture amazighe,”, Etudes et documents berbères, 27, pp. 159–172.

[14] F.Amounas. 2015. "Enhanced Elliptic Curve Encryption Approach of Amazigh alphabet with Braille representation", International Journal of Computer Science and Network Solutions, vol 3, No 8, pp. 1-9.

[15] F.Amounas and E.H. El Kinani. 2012. "Fast mapping method based on matrix approach for elliptic curve cryptography", International Journal of Information & Network Security, vol.1, No.2, pp. 54-59.

[16] Herbert Shildt. 2011, “Java complete reference”, Tata McGraw-Hill.