On enhancement of Elliptic Curve Encryption of Amazigh Text using Graph Theory

Fatima Amounas

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

E-mail: f_amounas@yahoo.fr

Abstract

Security in today's world is one of the important challenges. Data encryption is an important issue widely used to protect the data and to ensure the security. Graph theory is widely used as a tool of encryption, due to its various properties and its easy representation in computers as a matrix. This paper develops an enhanced version of elliptic curve encryption method using graph theory. It attempts to enhance the efficiency by providing add-on security to the ECC cryptosystem using graphs and its properties. The proposed method represents a novel approach to encrypt and decrypt Amazigh text securely with the benefits of graph theory properties.

Keywords: Elliptic Curve cryptography, Graph theory, Adjacency matrix, Unicode, Amazigh Alphabet.

I. Introduction

Cryptography is the combination of Mathematics and Computer science. Ever science several mathematical models were applied for encryption/ decryption. Cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break by any adversary. One of the mostly used in public key cryptography is the elliptic curve cryptography (ECC).

ECC depends on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP). So an adversaries are not able to solve ECDLP which is infeasible to be solved has a strength security against all kinds of attacks. ECC can be defined over two types of fields: first one is the prime field $F_p$ which is suitable for the software applications and the second one is the binary field which is suitable for the hardware applications. ECC offers the same security level like RSA and ElGamal cryptosystems with shorter key length (Lawrence C.Washington, 2008).

Recently, researchers developed several new encryption method based ECC. For instance, the authors in (Maria Celestin et al. 2013) proposed a new elliptic curve cryptosystem using decimal ASCII values to represent the characters. These characters are mapped into points on elliptic curve by multiplying their values by a random point which lies in the same curve. In our previous work (F.Amounas et al. 2013), we presented a novel approach based ECC to encrypt data using data matrix. In (V.Kamalakannan et al. 2015), the authors extend this approach and proposed the implementation of ECC with ElGamal cryptosystem for encryption and decryption a message.

Recently, Graph Theory is one such field which is being successfully integrated to provide stronger cryptographic algorithms. Graph theory is extensively used in encryption. In (Steve Lu et al. 2008), the authors use an arbitrary graph where every node and every edge are assigned an arbitrary image. Using this approach, pixel expansion and contrast are proportional to the number of images. In (M. Yamuna et al., 2013a) Hamiltonian circuit of a complete graph is
used as the tool for encryption and decryption of multiple messages. In (M. Yamuna et al., 2015), the author provided an encryption algorithm to encrypt and decrypt data securely with the features of graph theory. In (M. Yamuna et al., 2013b), the authors used a musical notes along with graph theory to encrypt binary messages. Here, the degree sequence of the graph constructed from any music note is used as the key.

Graph theory has a great contribution in the development of various encryption techniques. In this context, we propose a novel encryption scheme based ECC for enhancing the security of Amazigh text using graph theory.

The remainder of this paper is divided into four main sections. In the first, we give some basic details required in the proposed method. In the second section, we present the proposed method in detail. In the third section, we give our detailed report to this algorithm by utilizing an example. While in the fourth section, we make some conclusion and we try to suggest some future works.

II. Preliminaries

In this section we provide some basic details required in the proposed method.

A. Graph Theory

Graph theory is a branch of applied mathematics, which deals the problem with the help of graph. In mathematics and computer science, graph theory is the study of graphs, which are mathematical structures used to model pair-wise relations between objects (Narsingh Deo, 2004). A Graph G (V, U) consists of a set of objects, where V is the set of vertices and U is a set to edges that connect vertices each other. Let G be the graph with m edges and n vertices.

V is the set of vertices of the graph G:

\[ V = \{v_1, v_2, ..., v_n\} \]

U is the set of edges of the graph G:

\[ U \subset V \times V, U = \{e_1, e_2, ..., e_m\} \]

A walk from one vertex to another in which each vertex not appears more than once called a path. A cycle appears when the path starts from one vertex and return back to the same vertex, when the cycle consists of all vertices in the graph we called it a cycle graph. The graph called complete graph when there is an edge between any two vertexes in the graph.

Graphs represented in two main ways, adjacency-list and adjacency-matrix. The adjacency-list representation of graph G (V, E) consists of an array of V lists, one for each vertex in V. The adjacency-matrix of graph G is a symmetric binary matrix M= (m_{ij}) such that:

\[ m_{ij} = \begin{cases} 
1 & \text{if there is an edge between the } i^{th} \text{ vertex and the } j^{th} \text{ vertex.} \\
0 & \text{if there is no edge between them.}
\end{cases} \]

For properties of graph theory we refer to (Krishnappa 2009, Krishnappa et al. 2013).

B. Elliptic Curve

An elliptic curve E over a prime field \( F_p \) (\( p \neq 2,3 \)) is defined by

\[ E: y^2 = x^3 + ax + b \mod p \]

(1)

where \( a, b \in F_p \) and satisfy the condition \( 4a^3 + 27b^2 \neq 0 \). The set of all points \((x,y)\) that satisfy an elliptic curve Equation 1, with a special point \( \Omega \) (that is called a point at infinity), forms an elliptic curve group \( E(F_p) \).
**- Arithmetic on Elliptic Curve**

**- Point Addition**

Suppose M(x_M, y_M) and N(x_N, y_N), where M ≠ N, are two points lie on an elliptic curve E defined in Equation 1. The sum M + N results a third point R which is also lies on E. To add two points on E there are some cases on the coordinates of the points M and N as following:

- If M ≠ N with x_M ≠ x_N. Then, the sum of M and N is defined by:

\[
R = (x_R, y_R) \text{ where } \begin{align*}
    x_R &= s^2 - x_M - x_N \mod p \\
    y_R &= s(x_M - x_R) - y_M \mod p
\end{align*}
\]

with \( s = (y_N - y_M)/(x_N - x_M) \)

- If x_M = x_N but y_M ≠ y_N. Then, R = \( \Omega \).

**- Point doubling**

Let M (x_M, y_M) be a point lies on E.

Adding the point M to itself is called doubling point on an elliptic curve E.

\[ 2M = R (x_R, y_R) \]

where

\[
\begin{align*}
    x_R &= (s^2 - 2x_M) \mod p \\
    y_R &= (s(x_M - x_R) - y_M) \mod p
\end{align*}
\]

with \( s = (3x_M^2 + a)/(2y_M) \)

**- Scalar multiplication**

Let \( \alpha \) be an integer and P is a point lies on E. The scalar multiplication \( \alpha P \) can be computed using the point doubling and point addition operations. For more details on the theory of elliptic curves, we refer to (William S. 2011, Boruah D. et al. 2014).

C. Amazigh language

The, Amazigh language is a branch of the Afro-Asiatic (Hamito-Semitic) languages (Greenberg, 1966). Nowadays, it covers the Northern part of Africa which extends from the Red Sea to the Canary Isles and from the Niger in the Sahara to the Mediterranean Sea. In Morocco, this language is divided, due to historical, geographical and sociolinguistic factors, into three main regional varieties, depending on the area and the communities: Tarifite in North, Tamazight in Central Morocco and South-East, and Tachelhite in the South-West and the High Atlas. Amazigh language is recognized as an official language in Morocco in addition to Arabic.

Since 2003, the Amazigh language has its own writing system. This system is called Tifinaghe-IRCAM (Ameur et al., 2004). This system contains:

- 27 consonants including: \( \Theta, \chi, \xi, \lambda, \epsilon, \mu, \nu, \rho, \sigma, \alpha, \beta, \gamma, \delta, \theta, \zeta, \xi, \omicron, \upsilon, \phi, \chi, \psi, \omega \).
- 2 semi-consonants: \( \eta \) and \( \mu \).
- 4 vowels: three full vowels: \( \alpha, \epsilon, \delta \) and neutral vowel (or schwa) \( \emptyset \) which has a rather special status in Amazigh phonology.

No particular punctuation is known for Tifinagh. IRCAM has recommended the use of the international symbols.
Tifinagh is encoded in the Unicode range U+2D30 to U+2D7F (L. Zenkouar 2004). The list Tifinagh alphabet and the associated Unicode allocated by ISO are illustrated in (F.Amounas et al. 2012).

III. Proposed Method

The main idea of our contribution depends on using graph theory to generate the adjacency matrix of the complete weighted graph. Then, apply the basic algorithm of ECC to generate the strong cipher text. Figure 1 shows detailed design of encryption/decryption process.

Suppose A and B are two users wishing to communicate over insecure channel. Let us choose the user A as the sender who wants to encrypt and send a message to the user B (the receiver). Assume that user A and user B are agreed to use the elliptic curve:

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

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:

$$P_A = n_AP$$
$$P_B = n_BP$$

The secret key will be $K=n_A P_B = n_B P_A$.

- Encryption process

Step 1. Assign the Unicode value to each character of the message. Represent each value in hexadecimal form of two digits $(h_1,h_2)_{16}$. Then convert the hex code to decimal value.

Step 2. Imbed each character into point on elliptic curve using the encoding table.

Step 3. Add vertex for each character in the plain text to the graph $G$.

Step 4. Link the vertices together by adding an edge between each sequential character in the original message until we form a cycle graph.

Step 5. Weight each edge using the result values: $w_i P = P_{i+1} - P_i$ with $w_i P = P_1 - K$. Each edge’s weight represents the distance between the connected two vertices in the encoding table.

Step 6. Adding more edges to form a complete graph and create the adjacency matrix $M$.

Step 7. Compute $K=n_A P_B$ and Add a special character as starting character. Then, draw a weighted graph with the starting character.

Step 8. Find the minimum spanning tree (MST) and construct the corresponding adjacency matrix $M_1$. Then, by using addition and doubling points on elliptic curve, compute

$$Q_i = w_i P + K, \text{ i}=1, 2, ..., n.$$ 

with $w_i$ represents a weight value assigned to the edge in MST.

Step 9. Create the vector $V_1 = (Q_1, Q_2, ..., Q_n)$. Then, multiplying this vector by the adjacency matrix $M_1$ to get the encrypted vector $C = (C_1, C_2, ..., C_n)$.

Step 10. Transform $M$ to data matrix $PM$ with entries on elliptic curve. Then, send $(PM, C)$ as the cipher text.

- Decryption process

Step 1. After receiving the cipher-text, the receiver extracts a data matrix $PM$ and retrieves $M$.

Step 2. Convert the remaining blocks to a vector $C = (C_1, C_2, ..., C_n)$.

Step 3. Construct the complete weighted graph with the adjacency matrix $M$. Then, find the minimum spanning tree and construct the corresponding adjacency matrix $M_1$. 

4
**Step 4.** Compute $C \times M_1^{-1}$ to get $V = (Q_1, Q_2, \ldots, Q_n)$.

**Step 5.** Compute $K = n_B P_A$ and $w_i P = Q_i - K$, by using addition and doubling points on elliptic curve. The corresponding character indicates the starting character.

**Step 6.** Extract $w_i$ from $w_i P$, by solving the discrete logarithm problem.

**Step 7.** Draw a weighted graph with starting character. Here, $w_i$ is the weights assigned to the current edge.

**Step 8.** Compute the original message by decoding each vertex. Node $i$: $P_{i+1} = w_i P + P_i$ with $P_1 = w_1 P + K$. The result code represents $d_i$.

**Step 9.** Convert $d_i$ to hexadecimal form $(h_1 h_2)_{16}$ and find the match character from the encoding table.

---

**User A**

Create the encoding table using Unicode form.

Convert the plaintext to vertices. Add edges to link vertices to form cycle graph $G$.

Compute $K = n_A P_B$ that indicates the starting character of the graph $G$ and compute $w_i P = P_{i+1} - P_i$ with $P_1 = K$.

Assign the data value to all edges to construct the complete weighted graph and its adjacency matrix $M$.

Find the minimum spanning tree from the complete weighted graph with the special ch. and its adjacency matrix $M_1$.

Modify a matrix $M$ to $PM$ and multiply a vector $V_1 = (w_i P + K)$ with the adjacency matrix $M_1$ to get $C$.

---

**User B**

Gets the plaintext by decoding the result code using Unicode form.

Compute: $P_{i+1} = w_i P + P_i$ with $P_1 = K$. Then, extract $d_i$ and convert it to hexadecimal form.

Compute $V_1 = M_1^{-1} C$ using addition and doubling operation on elliptic curve. We get $w_i P = Q_i - K$.

Find the minimum spanning tree from the complete weighted graph with the special ch. and its adjacency matrix $M_1$.

Construct the complete weighted graph with its adjacency matrix is $M$. Then, add a starting character to the graph.

Compute the secure key $K = n_B P_A$ and find the starting character of the graph. Extract $M$ from a data matrix $PM$.
IV. Illustration with an example
In this section, we show the details of our encryption algorithm by an example. The elliptic curve using here is given by the following equation:

\[ E: y^2 = x^3 + 2x + 9 \mod 37. \]  

The graph of the function is shown in Figure 2.

Let the point (5, 25) be chosen as the base point P.

User A: choose the private key \( n_A = 19 \) and compute \( P_A = (23, 7) \).

User B: choose the private key \( n_B = 13 \) and compute \( P_B = (26, 5) \).

If user A wants to send the message “ΣΙΓΘ” to user B, he does the following:
The first step is to convert the plaintext to the graph, by converting each character to a vertex. Then, link two sequential characters together to form a cycle graph as shown in Figure 3.
Figure 3. Graph contains all characters of the original message.

The second step is to weight each edge using the encoding table (Table 1) as shown in Figure 4.

<table>
<thead>
<tr>
<th>character</th>
<th>Φ</th>
<th>Θ</th>
<th>...</th>
<th>ξ</th>
<th>ι</th>
<th>...</th>
<th>Ω</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hex code</td>
<td>30</td>
<td>31</td>
<td>...</td>
<td>49</td>
<td>4A</td>
<td>...</td>
<td>5B</td>
<td>4F</td>
</tr>
<tr>
<td>Dec. value</td>
<td>48</td>
<td>49</td>
<td>...</td>
<td>73</td>
<td>74</td>
<td>...</td>
<td>91</td>
<td>79</td>
</tr>
<tr>
<td>Point P&lt;sub&gt;i&lt;/sub&gt;</td>
<td>(25,12)</td>
<td>(4,28)</td>
<td>...</td>
<td>(21,5)</td>
<td>(0,34)</td>
<td>...</td>
<td>(13,30)</td>
<td>(5,12)</td>
</tr>
</tbody>
</table>

Next, draw an edge between every vertex pair to obtain a complete weighted graph. Each edge weight represents the difference between the corresponding points of the connected two vertices from the encoding table (Table 1).

Figure 4. Complete weighted Graph.

The result adjacency matrix is given as follows:

\[
M = \begin{pmatrix}
0 & 1 & 18 & 16 & 6 & 16 \\
1 & 0 & 17 & 15 & 5 & 6 \\
18 & 17 & 0 & -2 & -12 & 2 \\
16 & 15 & -2 & 0 & -10 & 0 \\
6 & 5 & -12 & -10 & 0 & 10 \\
0 & 7 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]
Now, create a data matrix PM with entries in elliptic curve. Here, the multiple of K are stored in the diagonal instead of 0 as follows:

\[
PM = \begin{pmatrix}
(13,7) & (5,25) & (35,21) & (31,22) & (27,5) & (31,22) \\
(5,25) & (1,30) & (26,32) & (21,5) & (4,28) & (27,5) \\
(35,21) & (26,32) & (21,32) & (16,20) & (13,7) & (27,32) \\
(31,22) & (21,5) & (16,20) & (7,25) & (9,33) & (0,1) \\
(27,5) & (4,28) & (13,7) & (9,33) & (25,12) & (33,14) \\
(31,22) & (27,5) & (27,32) & (0,1) & (33,14) & (4,28)
\end{pmatrix}
\]

After that, add a special character that points to the first character. It is the corresponding character of a secure key K. Figure 5 show the complete graph with the starting character.

![Figure 5. Complete weighted Graph with special character.](image)

Now, construct the corresponding adjacency matrix as follows:

\[
M_1 = \begin{pmatrix}
0 & -4 & 0 & 0 & 0 & 0 & 0 \\
4 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & -2 & -12 & 10 & 0 \\
0 & 0 & 0 & -2 & 0 & 0 & 0 \\
0 & 0 & 2 & -12 & 0 & 0 & 0 \\
0 & 0 & 0 & 10 & 0 & 0 & 0
\end{pmatrix}
\]

Then, create a data vector with entries are points on elliptic curve: \( V_1 = (Q_i), \) such that \( Q_i = w_i P + K, \) for \( i = 1, 2, \ldots \)
After that, multiply $V_1$ by the data matrix $M_1$ to form a vector $C$:

$$C = [(11,20), (31,15), (15,11), (10,20), (7,12), (29,6), (9,33)]$$

Now, send $C$ and $PM$ as the cipher text in a linear format. So, the encrypted message that is sent is:

```
1C3QAE9C6E7UPREILXYOR5H3E0X3R7E
```

For the decryption of the cipher text to the original message, the receiver B does the following steps:

The first step is to transform the encrypted characters to points on elliptic curve.

Next, extract a data matrix $M$ from the received data matrix $PM$. Then, we store the remaining points into the vector $C$ as follows:

$$C = [(11,20), (31,15), (15,11), (10,20), (7,12), (29,6), (9,33)]$$

Now, from the adjacency matrix, construct a complete weighted graph $G$.

After that, find the minimum spanning tree from the complete weighted graph with the special character $K$ and its adjacency matrix $M_1$.

The second step is to multiply a vector $C$ by the data matrix $M_1^{-1}$ to form a vector $V_1=(Q_i), i=1, 2, \ldots$ as follows:

$$V_1 = [(35,16), (7,25), (35,21), (29,31), (16,20), (33,23), (10,17)]$$

Then, compute the weight of each edge, using addition and doubling operation on elliptic curve:

$$w_i P = Q_i - K, i=1, 2, \ldots$$

Now, compute the embedding point that represents the current character and so on until we got the mapping points.

$$P_{i+1} = w_i P + P_i$$ with $P_1 = K$

Hence, reverse the embedding to get the original message: “Σίκωθ”. 

### V. Conclusion

Graph theory is growing as a promising field in various fields. In this paper we use graphs to enhance the security of the Amazigh text using elliptic curve. Here, the graph is assigned using addition and doubling of point on elliptic curve. For decrypting the message we need to know (i) the starting character as a secure key, (ii) The way the weights are assigned to all the edges. Decryption is almost impossible without the knowledge of this information. Here, it is difficult to find the encrypted graph and its properties, because it is difficult to solve the ECDLP. This encoding and decoding scheme of the proposed method is significantly different as compared to the traditional methods. This proposed method is hence a secure method for transmission of any sentence Amazigh. As a future work, the proposed method can be extended for encrypting the video messages as well as sound encryption process.
References


