Refined Imbricate Cryptography with addition of Polygram Substitution Cipher Method: an enhanced tool for security

M Dharshini¹, K Gayathri², S Renuga Devi³ and B Gopalakrishnan⁴

¹, ², ³ Post graduate in Software Engineering, Department Of Information Technology
⁴Associate Professor, Department Of Information Technology
Bannari Amman Institute of Technology, Sathyamangalam, India.
Email:dharshini.se19@bitsathy.ac.in

Abstract-The booming of technologies these days make us more comfortable and also take us into the complex world. As information increases rapidly, the level of security also matters. As a technological paradigm, the security can be done using many cryptography techniques. Cryptography is a process of converting a plain text into a cipher text so that the data can be viewed or extracted only by an intended user. In this paper, we describe a methodology inorder to enhance the speed of encryption and decryption process. This process uses ciphertext formation via Imbricate Cryptography and Polygram Substitution Cipher Technique. On performing cryptography, there are a lot of improvements in the actual implementation. Most importantly, the advantage of combining these two algorithms is to make data more secure from the exploiters and also making the computer to process faster.

1. Introduction
Protecting data in today’s world is not that easy because we are transferring the data through the untrusted medium called “Internet”. So there is a chance of getting threats from the hackers and the attackers. To keep data in a secret manner, we use cryptographic algorithms. By using some algorithms, the information needs to be coded into a readable format so that the intended user can only understand. This process is known as encryption and decryption. Encryption is a process of converting a data into a code by adding the key, especially to prevent unauthorized access. Decryption is a process of converting the data into the original plain text.

2. Literature Survey
Rohit Rastogi et al., proposed an encryption algorithm by establishing the relationship between Random and Cryptography. By using these techniques, they generated an algorithm based on the result of Linear Congruential Pseudo Random Number Generator with Imbricate Cryptography. In the first layer, they will implant a random number in the text. In the second layer, they will implant a key, a two way level. The original text can be retrieved only by knowing the corrected key and randomly generated number.

Ekta Agarwal et al., developed an algorithm in which it will first reverse the given plain text. Secondly, they will assign a key as much as they want. After applying the key, they will change into their equivalent ASCII code. Thirdly, they generate the binary value for that ASCII number and also do LSB. By adding the LSB value, the final result will be printed.
Aakash et al., proposed an encryption algorithm in which they encrypted a text with the Vigenere Cipher method. In this, polygram substitution cipher technique uses the Vigenere table for the encryption of characters. They also added some numbers and special characters so that any type of plain text can be encrypted.

Yassin Bouterra et al., proposed an algorithm based on the ASCII code. This paper is categorized as a polyalphabetic substitution symmetric cipher. In this, they will link the encrypted character with the previous characters and the same is done in the decryption method. This algorithm will encrypt the characters and numbers using the ASCII code.

Andini Dani Achmad et al., developed an algorithm in which they have used the Vigenere Cipher as a cryptographic technique that helps in securing the data. It generally works by shifting each character in the plain text for the provided key.

3. System Analysis

3.1 Problem Definition

This project has been developed on the motive of data security and also in increasing the processing speed by using the combination of Imbricate Cryptography and Polygram Substitution Cipher Method.

3.2 Existing Algorithm

3.2.1 Imbricated Cryptography

Imbricate Cryptography is an encryption tool that follows a layered approach. It uses a symmetric cryptography in which the key will be attached into the message so that the user can be accessed only by knowing the correct key. This algorithm consists of three layers of encryption.

3.3. Step 1: Mapping Layer

In this layer, each of the characters is replaced with the mapping character. It consists of two sets for frequent and occasional characters. Each and every character given in the plain text will be mapped with equivalent characters that present in the set proves the first step of encryption.

3.4. Step 2: Core-encoding Layer

In this layer, it uses bitwise logics and ASCII code format to encode the characters that come from the first layer. The first character of the message is negated with the ASCII character of the first character. This process is followed to the rest of the characters that present in the message.

3.5. Step 3: Converting to Bitmap

In this layer, it converts the ASCII characters into their equivalent binary value and stores it in a bitmap file.

Table 1. Table for Mapping

| Source File Characters | Mapping Characters |
|------------------------|--------------------|
| a,e,i,o,s,t { frequent } | o/t/s/i/a/e |
| b,c,d,e,f,g,h,j,k,l,m,n,o,p,q,r,u,v,w,x,y,z { occasional } | h,f,b,d,g,c,l,n,j,k,m,u,y,p,z,q,v,w,x,p |
| 0,1,2,3,4,5,6,7,8,9 { numerals } | 4,6,9,7,0,8,3,2,5 |
| Special Characters | Same Characters |
3.6. Polygram substitution Cipher Method.

In the traditional method, a block of characters will be altered by another block. As this technique faces many disadvantages, we have moved to the improvised Polygram substitution cipher method.

**Step 1:** Each block of characters should be reversed before applying to the Polygram substitution cipher method.

**Step 2:** Convert the characters to the ASCII form.

**Step 3:** If the block is repeated, the same process should be followed.

**Step 4:** ASCII Value should be added and subtracted.

3.6.1. Proposed Algorithm

This paper has been proposed for data security and to increase the processing speed by combining the Imbricate and Polygram substitution Cipher Method.

From this diagram, first we need a message and a key. Then, find the equivalent mapping character for that message. Convert the equivalent characters with the ASCII Codes. Subtract the key consecutively with the ASCII code. Then convert to the equivalent binary values. Apply LSB on the binary values. Finally, calculate the binary values. We will get a group of numbers as an encrypted message. Only if we found those numbers so that they can retrieve an original message. The working framework of decryption is reverse as an encryption mechanism.
4. Working Mechanism

4.1. Encryption Process
Step 1: Convert each character of the message with the equivalent mapping characters.
Step 2: Convert the characters that we get from the previous step to their equivalent ASCII Codes.
Step 3: Subtracting the key (i.e., converting each character of the key to their equivalent ASCII Codes) consecutively with the mapping character ASCII Code. Each character in the key will represent the number of keys. Step 4: Find the Binary Values of the subtracted values.
Step 5: Apply LSB for the Binary Values from the previous Step. Step 6: Finally, calculate the Binary Values (i.e., Encrypted Format.).

4.2. Decryption Process
Step 1: Take the ASCII value of each updated character and convert it to the binary values
Step 2: Perform LSB for the previous value that we get from the previous step.
Step 3: Converting the value with their equivalent ASCII Code. Step 4: Add the key consecutively with the ASCII Code.
Step 5: Write the character for the equivalent ASCII Code.
Step 6: Then, map with the equivalent characters to get the original plain text.

5. Results and Analysis
In this module, we are going to see how actually encryption and decryption works. Below, we are going to deal with some examples of how the refined Imbricate and Polygram substitution cipher method works.

**Plain Text:** Hello

**Key:** hai

5.1. Step 1: Mapping Characters

| Plain Text | Mapped Characters |
|------------|------------------|
| h          | l                |
| e          | t                |
| l          | j                |
| l          | j                |
| o          | i                |

Find ASCII Code for Key

| Key | ASCII Code |
|-----|------------|
| h   | 104        |
| a   | 97         |
| i   | 105        |
5.2. Step 2: Find the ASCII Code for each character in the message

| Mapped Char | ASCII Code |
|-------------|------------|
| l           | 108        |
| t           | 116        |
| j           | 106        |
| j           | 106        |
| i           | 105        |

5.3. Step 3: As the key as three characters h,a,i. ASCII Code for h,a and i are as follows k1-104, k2-97 and k3-105.

| Mapped Char | ASCII Code | Subtract Con. letter |
|-------------|------------|----------------------|
| l           | 108        | 4                    |
| t           | 116        | 19                   |
| j           | 106        | 1                    |
| j           | 106        | 2                    |
| i           | 105        | 8                    |

5.4. Step 4: Find the Binary Values

| Mapped Char | ASCII Code | Subtract Con. letter | Binary Values |
|-------------|------------|----------------------|---------------|
| l           | 108        | 4                    | 00000100      |
| t           | 116        | 19                   | 00010011      |
| j           | 106        | 1                    | 00000001      |
| j           | 106        | 2                    | 00000010      |
| i           | 105        | 8                    | 00001000      |
5.5. Step 5: Apply LSB for the binary values.

| Mapped Char | ASCII Code | Subtract Con. letter | Binary Values | LSB       |
|--------------|------------|----------------------|---------------|-----------|
| l            | 108        | 4                    | 00000100      | 00000101  |
| t            | 116        | 19                   | 00010011      | 00010011  |
| j            | 106        | 1                    | 00000001      | 00000000  |
| j            | 106        | 2                    | 00000010      | 00000011  |
| i            | 105        | 8                    | 00001000      | 00001001  |

5.6. Step 6: Calculate the Binary Values.

**Table 2. Encryption Table**

| Mapped Char | ASCII Code | Subtract Con. letter | Binary Values | LSB       | Final Result |
|--------------|------------|----------------------|---------------|-----------|--------------|
| l            | 108        | 4                    | 00000100      | 0000101   | 5            |
| t            | 116        | 19                   | 00010011      | 0001011   | 18           |
| j            | 106        | 1                    | 00000001      | 0000000   | 0            |
| j            | 106        | 2                    | 00000010      | 0000011   | 3            |
| i            | 105        | 8                    | 00001000      | 0000101   | 9            |

Encrypted Output: {5, 18, 0, 3, 9}

Decryption Process

**Table 3. Decryption Table**

| Encrypted Output | Binary Value | LSB       | ASCII Code | ADD Con. letter | ASCII Letter | Mapped Letter |
|------------------|-------------|-----------|------------|-----------------|--------------|---------------|
| 5                | 00000101    | 00000100  | 4          | 108             | l            | h             |
| 18               | 00010010    | 00010011  | 19         | 116             | t            | e             |
| 0                | 00000000    | 00000001  | 1          | 106             | j            | l             |
| 3                | 00000011    | 00000010  | 2          | 106             | j            | l             |
| 9                | 00001001    | 00001000  | 8          | 105             | i            | o             |
This graph represents the relationship between the Message and the Size of the Key. I have taken X-axis as Message and Y-axis as the size of the key so that we can calculate how much the length of the key will be generated for the particular message.

6. Conclusion and Future work
This Algorithm deals with the data security and also in the processing speed. It overcomes the disadvantages in securing the data. The proposed algorithm will help you in encrypting and decrypting the message by using the combination of Imbricate Cryptography and Polygram Substitution Cipher Method. By this, we can avoid threats from the hackers and the attackers. This is a strong encryption and decryption algorithm so that only the authorized user can decrypt and can retrieve the original message.

References
[1] Moditha Vasuki R, Pooja H D, Nikitha H M and A A Priyanka, Review on Improvised Imbricate Cryptography with Pseudo Random Number Generator using linear congruential Algorithm, National Conference on Technology for Rural Development (NCTFRD-18)
[2] Ekta Agarwal, Dr. ParshuRam.Pal, Refined Polygram Substitution Cipher Method - An enhanced tool for Security”, Internal Journal of Engineering and Innovative Technology (IJEIT), July 2012.
[3] Nalani N, Raghavendra Rao G Cryptanalysis of simplified Data Encryption Standard via Heuristic, IJCSNS.
[4] Aakash, Jitendra Kumar Soni, Bharti sharma, “A.J. Cipher”, 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), 2017.
[5] Rohit Rastogi, Shashank Mittal, Shashank Shekar, Linear Algorithm for Imbricate Cryptography using Pseudo Random Number Generator, 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom).
[6] N. Sharma, Prabhjot and H. Kaur, A Review of Information Security using Cryptography Technique.International Journal of Advanced Research in Computer Science,2017.
[7] N. Jirwan, A. Singh and S. Vijay, Review and Analysis of Cryptography Techniques, International Journal of Scientific and Engineering Research. 2013.
[8] S. Tayal, N. Gupta, P. Gupta, D. Goyal, A Review paper on Network Security and Cryptography, Advances in Computational Sciences and Technology, 2017.
[9] Ramesh. A, Tirunelvei, Performance analysis of encryption algorithms for Information Security, ICCPCT, March 2013.

[10] Deepak Kumar Dakate, Pawan Dubey, Performance Comparison of Symmetric Data Encryption Techniques, Internal Journal of Advanced Research in Computer Engineering and Technology, August 2012.