IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 4, Ver. III (Jul – Aug. 2014), PP 100-106
www.iosrjournals.org

Implementation and Result Analysis of Polyalphabetic Approach to Caesar Cipher

Prachi Patni[1]

[1] *(Computer Science & Engineering Department, Government College of Engineering, Aurangabad, India)*

Abstract: *In the modern world as there is drastic hike in use of internet for our daily work there is need to keep our information safe and secure so that an intruder can't misuse it. Cryptography was established to solve this problem. Cryptography is an art of transforming information (Plain Text) using encrypting algorithms into a form that is not readable (Cipher Text) without access to specific decoding algorithms.*

In this paper the author presents a novel approach to cryptographic techniques and illustrates the result and analysis of the proposed algorithm and points out that it is with improved security from many kind of attacks.

This paper is partitioned in following sections: 1st section contain basic introduction about cryptography and Caesar Cipher, 2nd section includes proposed system, 3rd contain performance analysis where proposed system is compared with other techniques, 4th include Conclusion and Future Scope and last section contains References.

Keywords: *Cryptography, Caesar, Cipher*

I. Introduction

Internet can be called as backbone of this modern era which includes interchanging of large amount of data between various communication channels.This modern era is dominated by paperless transactions in offices by means of E-mail messages, E-cash transactions, etc. In various business and commercial sectors, there may be some secret information like confidential information banking transactions, government information, credit information is transferred over web using social network, E-mails etc. So there is a need to develop a scheme that guarantee to protect the information from the attacker. Cryptology is at the way of providing such guarantee. The word cryptology is derived from two Greek words: kryptos, which means "secret or hidden" and logos, implies for "description". Cryptology comprises of two competing skills – concealment and solution. The concealment part of cryptology is known as cryptography. Cryptography is often called "code making." The solution part of cryptology is called cryptanalysis. Cryptanalysis is often called "code breaking". [12]

1.1 Cryptography

The purpose of cryptography is to convert plaintext (message) to unreadable cipher text.

1.1.1 Terminology used in cryptography:[6]

i) Plain Text: This is original message or actual secret message which person wants to send to other party.

ii) Cipher Text: This is encrypted message which is output of encryption algorithm. Cipher text message cannot be understood by intruder because of its non-readable format. Is can only be decrypted by authentic user by using key.

iii) Encryption Algorithm: This is the process of reconstructing plaintext into cipher text with use of key.

iv) Key: This is also given as an input to encryption algorithm. It may be alpha numeric, numeric or may be a special symbol.

v) Decryption Algorithm: This is the process of converting cipher text to pain text. It is a reverse method of encryption algorithm. Encryption algorithm takes place at the sender end and decryption algorithm takes place at the receiver end.

1.1.2 Goals of Cryptography

Cryptography renders lots of security goals to ensure the secrecy of data, unmodified data and so on. Following are the various goals of cryptography [13].

i) Authentication:
- It must be achievable for the recipient of a message to make sure of its source.
- An intruder should not be able to act as someone else.

ii) Integrity:
- It should be achievable for the recipient of a message to certify that the message has not been altered by an intruder.
- An intruder should not be able to switch a fake message for an original one.

iii) Non-Repudiation:

- The person who is sender/receiver of the message must not be able to deny later that he/she sent a message.

iv) Confidentiality:

- It ensures that information can only be understood by those who have permission to access the message.

1.1.3 There are three ways by which plain text can be converted to cipher text. [1]

i) Transposition technique: This technique includes rearranging of elements to change its appearance.

ii) Substitution technique: This technique includes replacing an element of plaintext into an element of ciphertext.

iii) Transposition- Substitution technique: In this both techniques are used.

The cryptography is divided into two main classes depending upon the security key they used for encryption and decryption of text.

1.1.4 Types of Cryptography

i) Symmetric Encryption: In symmetric key cryptography same key is used for encryption and decryption purpose i.e. key can be fetched from the decryption algorithm. The encryption algorithm produces the key and then sends it to receiver section where decryption takes place. It is much effective and faster than asymmetrical key cryptography [9]. Symmetric key encryption main drawback is that both the users need to transfer their keys in a secure way because if the key is compromised then whole system will be compromised.

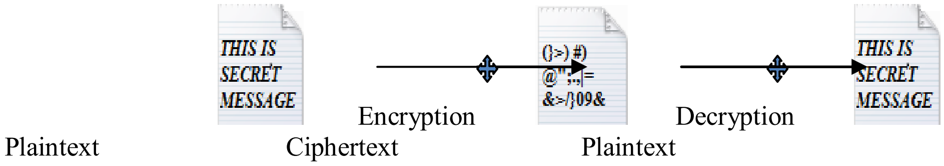

Figure 1. Symmetric Encryption

ii) Asymmetric Encryption: In Asymmetric key cryptography different keys are used for encryption and decryption. It is also called as public key cryptography. It consist of two keys namely public key and private key. Public key is known to the everyone and is used for encryption. Private key is known only the sender/reciever of that key and is used for decryption. The public key and the private key are correlated to each other by any mathematical means[9]. Figure 2 shows working of asymmetrical key cryptography.

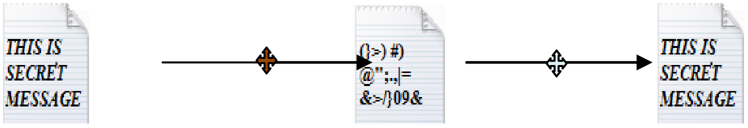

Figure 2. Asymmetric Encryption

1.1.5 Types of attacks on cryptography

i) Cipher text only attack (COA): In this type of attack it is assumed that only set of ciphertexts is available to the cryptanalyst. It is the weakest type of attack of all because cryptanalyst's lack of information.

ii) Known-plaintext attack**:** The attacker knows or can guess the plaintext for some parts of the cipher text. The attacker can now decrypt the rest of the cipher text blocks using this information. This can be accomplished by determining the key that was used during encryption of the data, or making use of some shortcut.

iii) Chosen-plaintext attack: The attacker is able to have any text he likes encrypted with the unknown key. The attacker requires determining the key used for encryption.

1.2 Caesar Cipher

Plaintext letters: zxcvqwertasdfgmnblkjhpoiuy
Ciphertext letters: JFOOFJJGFKJGLFMFGFFKKNKNKNV

Consider an example where a plaintext is given with its respective key. If anybody wants to remember that key it is nearly impossible, so other option is to write the key but it can also be lost, stolen or forgotten. It is necessary to create a key which need not to be written [1]. Therefore various algorithms were proposed to solve this problem. Caesar cipher is one among them. It is called by many names like shift cipher, Caesar's code or Caesar shift. It is one of the simplest and most widely known classical encryption techniques. It is an example of a substitution cipher method [13]. It was used by Julius Caesar to communicate with his army. Caesar decided that he would be shifting each letter by three places left, and so he informed all of his generals of his decision, and then he sent them that encrypted messages [13]. One of the strengths of the Caesar cipher is its ease of use.

It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter by shifting some positions. For example, suppose shift is 3, A would become D, B would be replaced by E, and so on. As with all Monoalphabetic substitution ciphers, the Caesar cipher can be easily broken [1]. The encryption can also be act as a substitute for modular arithmetic by first replacing the letters by numbers, according to the scheme, A=0, B=1… Z=25.

Encryption of a letter x by a shift n can be representing mathematically as:

$$E_n(x) = (x + n) \, mod \, 26 \qquad (1)$$

Similarly, decryption can be represented as follows

$$D_n(x) = (x - n) \, mod \, 26 \qquad (2)$$

The key can be remembered easily because there is of the pattern in it. Sender and receiver just needed to remember the shift.

II. Proposed Algorithm

In this section, the main algorithm which is used in the proposed system is described. The proposed system, which is used to encrypt the text, is divided into the following 3 main phases:

Phase-1: Creating Ciphertext1: In this phase we open a file and read it stream by stream. The stream is converted into block of 8. Now the block is XORed with array of key generated by Random Number Generator. Now the block created will be used as key for the second stream.

Phase-2: Applying Caesar Cipher: The Ciphertext1 is encrypted again using algorithm Caesar Cipher.

Phase-3: Creating Ciphertext2.

Each above phase consists of number of sub-phases. The detailed description for each of above phase is described as follows.

The flow for proposed system is as depicted in Figure 3

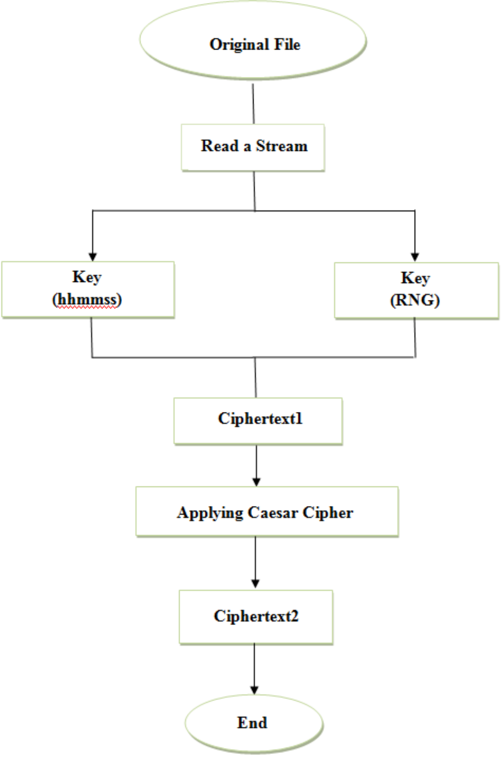

Figure 3: Flow Diagram for Proposed System

This proposed algorithm we created two Ciphertexts because if the file contains two same words they should be encrypted differently. This algorithm also contains two keys. The first reason for having two keys is that this will helps in increasing security. As we know that if the security is compromised then the whole system will be compromised. So it is most necessary and important to make security of key as unbreakable as possible.

PHASE 1:

Step 1: The first Step is reading text stream by stream from a file. Stream must be converted into block of 8. So length of the first stream is calculated.

- If the length of the string is greater than 8, 8 characters from the starting is taken to create a block.
- If the length of the string is found to be less than 8 then padding by 0 is done.

In this way a string is created of size 8.

For example if the first string is "hello" then it will become "hello000" for converting it into ciphertext1. Else if the string is "encrypting" then it will take only "encrypti"

Step 2: Keys are created

- Key 1: The first key is created by using "time" and "date".

For e.g. if today's date is: 13-06-2014 and time is 22: 56:54 so the required key will be 22565420140613. So if the intruder has plaintext and algorithm and try's to create a new file even after 1 second then also the text will be change and the authentic user will know that the file is modified.

- Key 2: The user will enter a random number and according to that number a Random Number Generator will create a number Between 1 to 1000.

For e.g. if user enter 34 as a random number then the Random Number Generator which is an inbuilt function in C# will produce a number say 991.

Both the keys are encrypted in different pattern so that if the intruder can able to guess the first key he/she will not able to compromise the system because the second key will protect the integrity of the system. After encrypting the first and second key the key is written to the output file.

Step 3: Random array is XORed with block of plaintext

By using key1 and key2 as input to Random Number Generator an array of 8 numbers is created. For example [1, 2, 3, 4, 3, 2, 0]. If the block is first block of the file then the string is XORed with random array generated let give name as "A". "A" must not exceed from 25 so for each ciphertext1 which will behave like a key for another blocks they are divided by 25. For second block "A" will be XORed with the second block let give name it as "B". "B" is divided by 25. Then "B" will be XORed with third block and so on.

PHASE-2: Applying Caesar Cipher

Step 4: The Ciphertext1 is again converted into the real length of string.

Step 5: Generate timekey for the Caesar Cipher

In this step timekey for the Caesar cipher is generated. The timekey is generated by the KEY1 by summing each and every digit of Key1.

For e.g. if Key1 is 22565420140613 then the timekey created for Caesar cipher is 2+2+5+6+5+4+2+0+1+4+0+6+1+3 = 41.

Step 6: Generation of Secret key by timekey + Key2 + String Length for Caesar Cipher.

In this step the secret key for the Caesar cipher is created. The Secret key is created by combining three parts. The first part is timekey which is created in the previous Step (Step 5). The second part is Key2 which is created in phase 1 and the third part if finding the length of the string. So the Secret Key will be sum of timkey, Key1, String Length

Step7: Split the string into odd and even position characters.

As the step is clear itself. The string is divided into two arrays. The first array is of the characters which are placed on the odd position in the string and second array is of characters which are placed on the even position in the string.

Step 8: Reverse odd and even position characters.

Step 9: Apply Caesar cipher on odd position characters from right to left and decrease key by 1.

Step 10: Apply Caesar cipher on even position characters from left to right and decrease key by 1.

Step 11: Merge even and odd position characters.

PHASE 3: Creating Final Ciphertext.

Step 12: From the string generated in the previous step file 1st middle and last character and subtract them by 26.

Step13: Repeat all the steps for each and every word till all the word are converted into ciphertext2.

Step 14: End.

The decryption algorithm will work in reverse order of the encryption algorithm. First step is same for encryption as well as decryption i.e. reading the text string by string. Secondly 1st middle and last character is found then they are added with 26. Then split the ciphertext in even and odd position arrays. Applying Caesar cipher to both the arrays. Generate plaintext1 and with the help of Key1 and Key2 generate final plaintext.

III. Performance Analysis

In this paper, the popular algorithms including AES (Rijndael), DES, 3DES were implemented, and their performance was compared by encrypting input files of varying contents and sizes. The algorithms were implemented in C#.NET.

Table1 and Table2 show the comparison between various algorithms [7].

Table 1: Comparison between various encryption algorithms

Input Size (in Kbytes)	AES	DES	3DES	Proposed Method
49	56	29	54	45
59	38	33	48	47
100	90	49	81	55
247	112	47	111	63
321	164	82	167	74
694	210	144	226	100
899	258	240	299	122
963	208	250	283	150

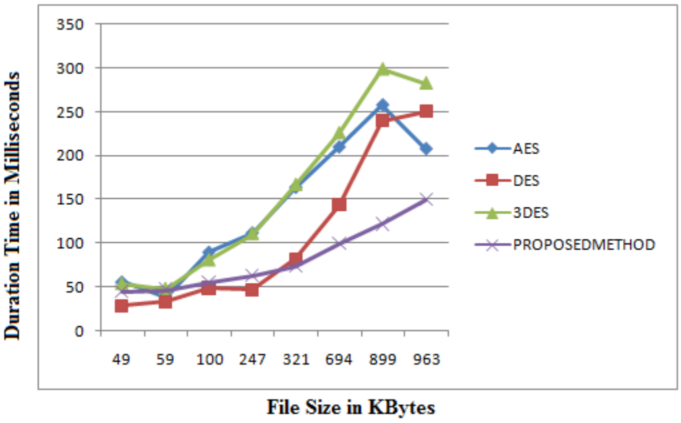

Figure 4: Graph showing encryption time for different file size

Table 2: Comparison between various decryption algorithms

Input Size (In Kbytes)	AES	DES	3DES	Proposed Method
49	63	50	53	50
59	58	42	51	52
100	60	57	57	58
247	76	72	77	65
321	149	74	87	76
694	142	120	147	103
899	171	152	171	120
963	164	157	177	130

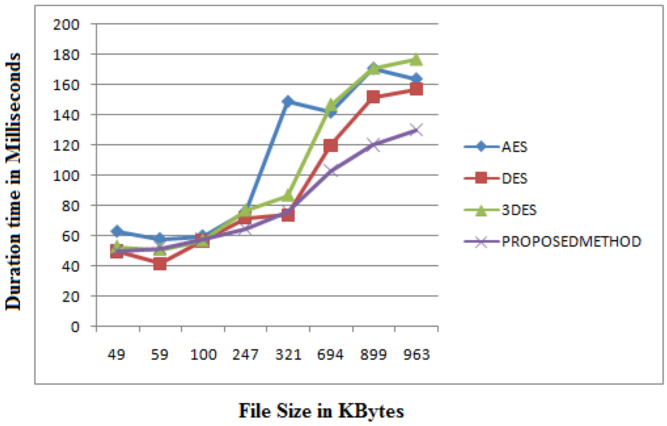

Figure 5: Graph showing encryption time for different file size

The task of performance analysis of the proposed system is accomplished by executing some experimental testing. For finding the performance of the proposed method some factors are considered. By comparing these factors with the other algorithms analysis is done

1. Tunability: It could be advantageous to be able to dynamically specify the encrypted portion and the encryption parameters correspond to different requirements. In this paper, tunability can have only one value either 'yes' or 'no'.

2. Computational Speed: In many applications it is required that encryption process as well decryption process to be fast
3. Key Length Value: In the encryption process the key management is the important aspect that shows how the data is encrypted. The symmetric key algorithm uses a variable key length therefore they use longer key. Hence, there is a need of key management.
4. Encryption Ratio The encryption ratio is the measure of the amount of data that is to be encrypted. Encryption ratio must be small to reduce the complexity.

Table 3 shows comparison between various techniques on different factors [8]

Table 3: Performance Comparison of Image Encryption Schemes

Algorithm	PlainText	CipherText	Key	Encryption Ratio	Speed	Tunability	Security Against Attacks	Advantages
AES	128	128	128 /192 /256	High	Fast	No	1) Chosen- plain text 2) Known-plain text	1) Reliable 2) Longer Key length supported
DES	64	64	56	High	Fast	No	1) Brute-force attack	1) Less no of computation. 2) Simple and fast. 3) Cryptanalysis is difficult
3DES	64	64	168	Moderate	Fast	No	1) Chosen- plain text 2) Known-plain text Brute-force attack	1) More Reliable 2) Longer key length 3) Cryptanalysis is difficult
BLOWFISH	64	64	32-448	High	Fast	Yes	1) Dictionary attacks	1) Fast and secure. 2) Compact.
RSA	MIN. 512	MIN. 512	512-1024	High	Fast	No	1) Timing attacks	1) Offering high performance 2) Delivering broad platform support
Proposed System	64	64	112-168	Moderate	Fast	No	1) Chosen - plain text 2) Known - plain text 3) Chosen - cipher text	1) Fast and secure 2) Contain two keys 3) Less no. of computations.. 4) Cryptanalysis is difficult

IV. Conclusions And Future Scope

In this paper, the author proposed and implemented an approach for text encryption and decryption. Proposed technique for secure transmission of Plaintext file has covered many limitations and security threats of older techniques of similar kind. Increasing levels of security provided in proposed algorithm including the block transformation, XOR technique, randomized bits and use of modified Caesar cipher. The algorithm itself has many advantages for increasing security. The use of two key gives double security. The polyalphabetic approach to Caesar cipher makes proposed algorithm both secure and easy to implement. The use of different key generation algorithm secure proposed application from various attacks.

From the implementation and testing results it is concluded that the proposed method is suitable for many types of text files like Excel, text, and word document. The algorithm is independent of machine i.e. if the other system has the application installed then he/she can encrypt the file and send encrypted file via internet and receiver can decrypt it. The computational speed of algorithms depends on the size of the text file to be encrypted or decrypted. Another main feature of this method is that it satisfies the properties of Shannon's Confusion and diffusion and Kerckoff's law so without knowing keys it makes decryption nearly impossible

Future Scope

- Security of the key can be improved.
- Encryption time and Decryption time can be decreased.
- In case of Caesar cipher any other algorithm can be used

References

[1]. Prachi Patni, A Poly-alphabetic approach to Caesar Cipher Algorithm, International Journal of Computer Science and Information Technologies (IJCSIT), Volume 4, 2013, 954-959, ISSN: 0975-9646.

[2]. Amit Joshi and Bhavesh Joshi, A Randomized Approach for Cryptography in Emerging Trends in Networks and Computer Communications (ETNCC), 22-24 April 2011.

[3]. S G Srikantaswamy and Dr. H D Phaneendra, Improved Caesar Cipher with Random Number Generation Technique and Multistage Encryption, International Journal on Cryptography and Information Security (IJCIS), Vol.2, No.4, December 2012.

[4]. Ramandeep Sharma, Richa Sharma and Harmanjit Singh, "Classical Encryption Techniques" published in International Journal of Computers & Technology, Volume 3. No. 1, AUG, 2012.

[5]. O.P. Verma, Ritu Agarwal, Dhiraj Dafouti and Shobha Tyagi, Performance Analysis of Data Encryption Algorithms, IEEE Delhi Technological University, India, 2011.

[6]. Somdip Dey, SD-AREE: An Advanced Modified Caesar Cipher Method to Exclude Repetition from a Message published in International Journal of Information & Network Security (IJINS), Vol.1, No.2, June 2012, pp. 67~76.

[7]. Diaa Salama Abd Elminaam, Hatem Mohamed Abdual Kader, and Mohiy Mohamed Hadhoud, Evaluating The Performance of Symmetric Encryption Algorithms, International Journal of Network Security, Vol.10, No.3, PP.213-219, May 2010.

[8]. AL. Jeeva, Dr. V. Palanisamy, K. Kanagaram, Comparative Analysis Of Performance Efficiency And Security Measures Of Some Encryption Algorithms, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 2, Issue 3, May-Jun 2012, pp.3033-3037.

[9]. Kashish Goyal and Supriya Kinger, Modified Caesar Cipher for Better Security Enhancement published in International Journal of Computer Applications (0975 – 8887)(IJCA), Volume 73– No.3, July 2013.

[10]. William Stallings, Cryptography and Network Security, Fourth Edition (Prentice-Hall) pp.80-81.

[11]. http://www.cs.trincoll.edu /~crypto/historical/caesar.html (Savarese, C and Hart, B, The Caesar Cipher, Last updated: 04/26/2010 03:46:57).

[12]. http://www.nku.edu (Fall 2006 Chris Christensen).

[13]. CRYPTOGRAPHY, https://en//.wikipedia.org/wiki/cryptography.