Volume 5, Issue 11, November 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Implementation of A Modified Version of Caesar Cipher Algorithm Using Prime Number

G. Sasikala, D. Aruna
Assistant Professor
Department Of Computer Science & Applications
Adhiparasakthi College Of Arts And Science(Autonomous)
Vellore District
Tamil Nadu

Abstract— Communication, Sharing, and Distribution are the terms which attract unauthorized users to attack communication among networked user. So security mechanisms are needed to protect original data through transmission. The security mechanisms are based on basic classical encryption techniques like Caesar cipher, play fair, Hill cipher, poly Alphabetic. This paper covers different kinds of security based algorithms on encryption techniques, which are most useful in the implementation of secured network. The objective is to implement new encryption algorithm of Caesar cipher substitution technique which overcomes the problems of existing ones by introducing prime number generation as a dynamic key, and add the positioning value of each alphabet in the plaintext. Once converted, it is unbreakable by intruders.

Keywords— Cryptography, Encryption, Decryption, Cryptanalysis, Intruders.

I. INTRODUCTION

The introduction of distributed systems and the use of networks and communication facilities for carrying data between users, affect the computer security. Network security measures are needed to protect data during communication. For an example, simply "password protection" is enough to secure a computer. In case of Network Security, so much of mechanisms and algorithms are needed to protect the communication and sharing of data from unauthorized users attacks. There is no single mechanism that will provide all the services like Identification, Authorization, Validation, Registration, Approval and disapproval etc.,. There is one specific element is available in most of the security mechanisms in use: **Cryptographic Techniques**. Encryption or Encryption like transformations of information is the most common means of providing security. This Paper focuses on the management of such techniques.

1.1 CRYPTOGRAPHY

Cryptography is the study of design of encryption techniques. The following diagram depicts the conventional encryption model as:

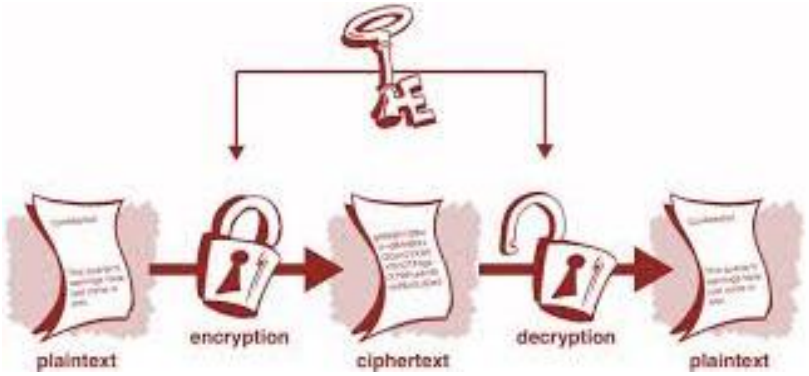


Figure 1.1.1 Conventional Encryption Model

The Terminologies used in Encryption Model are:
Plain Text : The actual information which the sender wants to transmit. It is readable text.
Cipher Text : the converted text which is in unreadable format.
Encryption : The process of converting plain text into cipher Text.
Decryption : The process of converting cipher text into plain text.
Secret Key : The confidential data between sender and receiver. It is also input to the algorithm. Encryption and decryption algorithm depend on this key.

1.2 CRYPTANALYSIS

Cryptanalysis is the process of discovering the plain text and it deals with the concept of defeating cryptography. One possible attack is assuming encryption algorithm and do brute force approach of trying all possible keys. If the key space is very large then this becomes impractical. These fundamental concepts may be helpful to pursue the depth knowledge of encryption techniques.

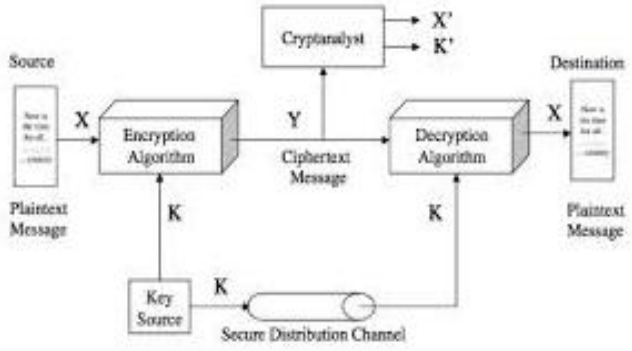


Figure 1.2.1 Cryptanalysis

1.3 ENCRYPTION TECHNIQUES

Generally encryption techniques are of two types. They are

i) Substitution Techniques,
ii) Transposition Techniques.

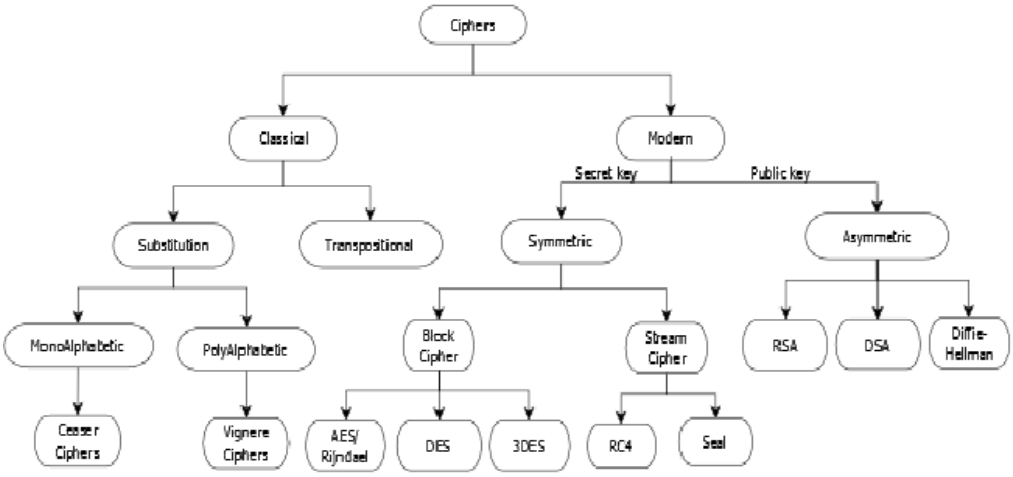


Figure 1.3.1 Classification of Encryption Techniques

A substitution cipher is one in which each character in the plaintext is substituted for another character in the cipher text. A Transposition Cipher is a cipher in which the plaintext message is rearranged by some means agreed upon by the sender and receiver.

1.4 CAESAR CIPHER

In the 19th century Julius Caesar was introduced simple substitution technique, called as Caesar cipher. It is the origin of all encryption techniques. In Caesar cipher each letter in the plain text is shifted 3 places ahead of it.

For example:

plain: meet me after the toga party

cipher: PHHW PH DIWHU WKH WRJD SDUWB

Note that the alphabet is wrapped around, so that the letter following Z is A. We can define the transformation by listing all possibilities, as follows:

Plain: a b c d e f g h i j k l m n o p q r s t u v w x y z

cipher: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Let us assign a numerical equivalent to each letter:

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12
n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

1.4.1 Encryption Algorithm

Encryption algorithm of Caesar cipher is expressed as follows. For each plaintext letter *p*, substitute the

Cipher text letter *C*:

A shift may be of any amount, so that the general Caesar algorithm is:

$$C = E(p) = (p + k) \bmod (26)$$

where *k* takes on a value in the range 1 to 25.

1.4.2 Decryption Algorithm

The decryption algorithm is simply:

$$p = D(C) = (C - k) \bmod (26)$$

1.4.3 Merits & Demerits

If it is known that a given cipher text is a Caesar cipher, then a brute-force cryptanalysis is easily performed: simply try all the 25 possible keys.

Three important characteristics of this problem enabled us to use a brute-force cryptanalysis:

- The encryption and decryption algorithms are known.
- There are only 25 keys to try.
- The language of the plaintext is known and easily recognizable.

In most networking situations, we can assume that the algorithms are known. What generally makes brute-force cryptanalysis impractical is the use of an algorithm that employs a large number of keys

II. RELATED WORK

In this section, the various methodologies for the encryption techniques used by various papers are provided:

Programmer Enas Ismael Imran [1] et.al proposes three different methods for both encryption and decryption. In Method1, the plain text/ cipher text correspondence for Caesar cipher depends on address of message, if the message contains an address, we just account the number of words in address of message, then apply Caesar cipher to encrypt the plain text by shifting the key value depend on the number of words in address. If we know the key we can decrypt the cipher text to plain text. In Method2, the key depends on the number of characters in the first word in plain text. Then apply Caesar cipher to encrypt the plain text based on the key. In Method3 defines by getting the original message and apply Caesar cipher to encrypt the plain text by shifting the key depend on number of words in the given plain text. We have shown that Caesar cipher being one of the simplest and widely used encryption techniques can be fortified beyond what common Caesar cipher can achieve.

Dr.A.Padmapriya [2] et.al provides the inverse of Caesar cipher more security for the data compared with earliest Caesar cipher. The algorithm gets the plain text and key value from the user. The key value should be in the range of 0 to 256. Though many solutions have been proposed most of them only consider 26 alphabets only. The main scope of this paper is the new level of data security solutions with encryption using ASCII full characters which is important for designing the complete security solution.

Ochoche Abraham [3] et.al proposes Improved Caesar Cipher(ICC) strengthen cipher text generated by Caesar cipher by removing spaces in plain text and changing the position of characters in original. Caesar cipher text to generate a new ICC cipher text that cannot be decrypted without integer values representing space locations in plain text, encryption key and the ICC algorithm.

III. PROPOSED ALGORITHM

In contrast to the Caesar cipher algorithm, our proposed algorithm generates a dynamic key from plain text in two phases.

Phase 1: Prime Number (Key) Generation

Phase 2: Incrementing Position value for each character in plain text

In Phase1, find the length of the text, generate the prime number and choose the largest prime number and consider it as secret key. In phase2, the ASCII value of each character is added with the prime number and its positioning value, do the reverse process in decryption.

3.1 ALGORITHM: ENCRYPTION

3.1.1 PHASE-I

Step 1: Get the plain text from the user to encrypt.

Consider an example,

Plain Text = "hi welcome"

Step 2: Calculate the length of the plain text including spaces.

Pain Text = "hi welcome"

Length = 10

Step 3: For the secret key, find prime numbers between 1 to length of the string.

Prime numbers = 2,3,5,7

Step 4: Choose the largest prime number from the set of prime numbers found.

Prime number = 7 (7 is taken as secret key)

3.1.2 PHASE-II

Step 1:

$$E(x) = \sum(x, p, i) \bmod 256$$

Where x = plain text,
p = prime number
i = iteration value

The ASCII value of each character is added with prime number and its positioning value.

3.1.3 Implementation of Encryption:

Plain Text	**h**	**i**	**space**	**w**	**e**	**l**	**c**	**o**	**m**	**e**
ASCII	104	105	32	119	101	108	99	111	109	101
Prime	7	7	7	7	7	7	7	7	7	7
Position	0	1	2	3	4	5	6	7	8	9
Key	7	8	9	10	11	12	13	14	15	16
Result=ASCII+ Key	111	113	41	129	112	120	112	125	124	117
Cipher text	**o**	**q**	**)**	**ü**	**p**	**x**	**p**	**}**	**|**	**u**

3.2 ALGORITHM: DECRYPTION

3.2.1PHASE – I

Step 1: Take the cipher text to decrypt,
Consider cipher text = oq)üpxp}|u

Step 2: Calculate the length of the string
Cipher Text = oq)üpxp}|u
Length = 10

Step 3: For the Secret key, find prime numbers between 1 to length of the text.
Prime numbers = 2,3,5,7

Step 4: Choose the largest prime number from the set of prime numbers found.
Prime number = 7 (7 is taken as secret key)

3.2.2 PHASE – II

Step 1:

D(x) = x- p-i mod 256

Where x = cipher text,
p = prime number
i= iteration value

The prime number/secret key and positioning value will be subtracted from the ASCII value of each character.

3.2.3 Implementation Of Decryption:

Cipher text	**o**	**q**	**)**	**ü**	**p**	**x**	**p**	**}**	**|**	**u**
ASCII	111	113	41	129	112	120	112	125	124	117
Prime	7	7	7	7	7	7	7	7	7	7
Position	0	1	2	3	4	5	6	7	8	9
Key = prime number + Position	7	8	9	10	11	12	13	14	15	16
Result = ASCII – Key	104	105	32	119	101	108	99	111	109	101
Plain Text	**h**	**i**	**space**	**w**	**e**	**l**	**c**	**o**	**m**	**e**

In Above plain text “hi welcome” the letter ‘e’ is repeated twice but two different cipher texts is generated and also for different alphabets same cipher text is generated. (For e & c alphabets in plaintext the letter p is generated)

The one more highlights in this proposed system is, it considers all the special symbols including space. In above example the plain text having space in between two words is considered.

This makes so complicated to guess the plain text by the un authorized person or intruders. Brute Force attack is not enough to break the plain text.

IV. CODING

```
Dim alp1(256)
 Dim alp2(256) As Integer
Private Sub Command1_Click()
Dim s As String
Dim op As String
Dim n As Integer
Dim l As Integer
Dim pos(500) As Integer
s = Text1.Text
l = Len(s)
n = prime(l - 1)
```



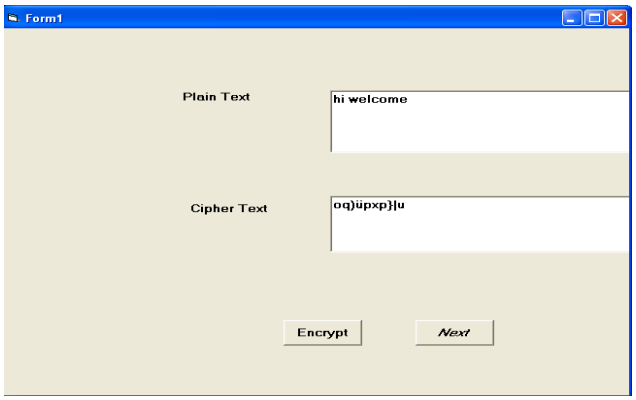
```
s = LCase(s)
For i = 1 To l - 1
pos(i) = position(Mid(s, i, 1))
pos(i) = pos(i) + n + i
pos(i) = pos(i) Mod 256
op = op + alpha(pos(i))
Next i
Text2.Text = op
End Sub
Function prime(n As Integer)
Dim d As Integer
Dim i As Integer
Dim j As Integer
Dim k As Integer
k = 1
Dim a(500) As Integer

For i = 2 To n
d = 1
  For j = 2 To i - 1
    If i Mod j = 0 Then
       d = 0
       GoTo l
       End If
    Next j
l:  If d = 1 Then
    a(k) = i
    k = k + 1
  End If
Next i
d = a(1)
For i = 2 To k - 1
If d < a(i) Then
d = a(i)
End If
Next i
prime = d
End Function
Function position(c As String)
Dim i As Integer
For i = 0 To 255
If c = alp1(i) Then
position = alp2(i)
End If
Next i
End Function

Function alpha(a As Integer)
Dim i As Integer
For i = 0 To 255
If a = alp2(i) Then
alpha = alp1(i)
End If
Next i
End Function

Private Sub Command2_Click()
Form2.Show
End Sub

Private Sub Form_Load()
 For i = 0 To 255
  alp1(i) = Chr(j)
```



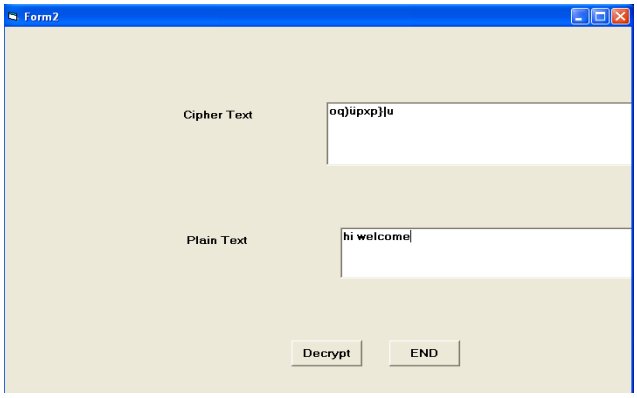
```
alp2(i) = i
j = j + 1
Next i
End Sub
```

V. SCREEN SHOTS

ENCRYPTION:



DECRYPTION:



VI. RESULTS AND DISCUSSION

This Section presents performance and comparison with respect to various parameters.

Parameters	**Caesar Cipher Algorithm**	**Enhancement Caesar Cipher for better security**	**Cloud Computing: Reverse Caesar Cipher Algorithm to Increase Data Security**	**Improved Caesar Cipher Algorithm**	**Proposed Algorithm**
Number of methods	One method	Three methods	Two methods	Two methods	Two methods
Keys Used	Adding fixed number with each alphabet	Adding fixed number with each alphabet depending on the address of the message, length of first word in message, number of words in first line	Use the key value from the range 0 to 256.	Adding fixed number to each alphabet and remove spaces between words	Generating Prime number and Adding the positioning value with each alphabet.
Cipher Text	Alphabets	Alphabets	Alphabets, numbers, symbols	Alphabets, spaces	Alphabets, numbers, symbols, spaces
Disclosure of cipher text	Easy	Moderate	Complex	Complex	Very complex
ASCII table used	No	No	Yes	No	Yes
Key Length	26	26	256	256	256
Security Against Attack	Brute Force	Brute Force	Predictable	Predictable	Unpredictable

VII. CONCLUSION

The proposed algorithm is an improvement over traditional substitution cipher algorithm. Rather than utilizing the 26 alphabets stated in earlier Caesar cipher algorithm, we have introduced 256 set of ASCII characters. In Caesar cipher, the key value is fixed, so we can break the consistent use of the same number as key for each letter. Instead of that we find prime numbers based on the length of plain text and choose largest prime number. To make more complicate, the key value is calculated with the addition of positioning number of each character in plain text along with chosen prime number. So that it is more dynamic and variable. It means each character get unique key value, and also not having the problem of frequency of the alphabet (i.e., same letter has different key value) which makes more secure and unpredictable to break.

REFERENCE

[1] Programmer Enas Ismael Imran, Programmer Farah abdulameer abdulkareem "Enhancement Caesar Cipher for Better Security", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. V (May-Jun. 2014), PP 01-05

[2] Dr.A.Padmapriya, M.C.A., M.Phil., Ph.D, P.Subhasri, (M.Phil, Research Scholar) ," Cloud Computing: Reverse Caesar Cipher Algorithm to Increase Data Security" International Journal of Engineering Trends and Technology (IJETT) - Volume4Issue4- April 2013 ISSN: 2231-5381

[3] Ochoche Abraham, Ganiyu O. Shefiu "AN IMPROVED CAESAR CIPHER (ICC) ALGORITHM" [IJESAT] INTERNATIONAL JOURNAL OF ENGINEERING SCIENCE & ADVANCED TECHNOLOGY Volume-2, Issue-5, 1198 – 1202 ISSN: 2250–3676

[4] Anupama Mishra, ENHANCING SECURITY OF CAESAR CIPHER USING DIFFERENT METHODS, IJRET: International Journal of Research in Engineering and Technology, eISSN: 2319-1163 | pISSN: 2321-7308,

[5] 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 DOI:10.5121/ijcis.2012.2405 39

