Journal of Academic and Applied Studies
Vol. 3(6) June 2013, pp. 1-7
Available online @ www.academians.org

ISSN1925-931X

EABC: Data Encryption Method Based on Circle

Hamid Mehdi

Department of Computer, Andimeshk Branch, Islamic Azad University,
Andimeshk, Iran

Abstract

Nowadays, there are many encryption algorithms to protect information from abuse. Data confidentiality is one of the most important functions of encryption algorithms, it means when the transferring data between different systems is vague for unauthorized systems or people. Moreover Encryption algorithms must maintain data integrity and provide availability for information. New encryption methods cause the attackers can not simply access to the information and do not allow discovering the relationship between information and the encrypted one. Therefore availability can be difficult for them. Existing complexities make their longevity and effectiveness increase (Mandal, 2012).In This Article, It has been tried to present an encryption method which has the characteristic of encryption algorithms and also has some unique complexities which are not easily detectable and efficient.

Keywords: Encryption algorithm, decryption, feistel, EABC algorithm.

I. Introduction

Encryption algorithms are divided into two parts (Shanta, 2012) the first public-key algorithms such as RSA. The Second private key, this part is divided into two categories in turn. The first subbranch is stream ciphers and the second subbranch is blocks ciphers. Renowned algorithms such as DES (Shanta, 2012), 3DES (Pavithra et al. 2012), AES (Thakur et al. 2011), Blowfish (Agrawal et al. 2010) can be mentioned as some instances for Block ciphers.

Encryption algorithms have advantages and disadvantages but what is of great important is, they are beneficial. The Main differences of encryption algorithms are computational algorithm in computation time, memory consumption and output bytes.

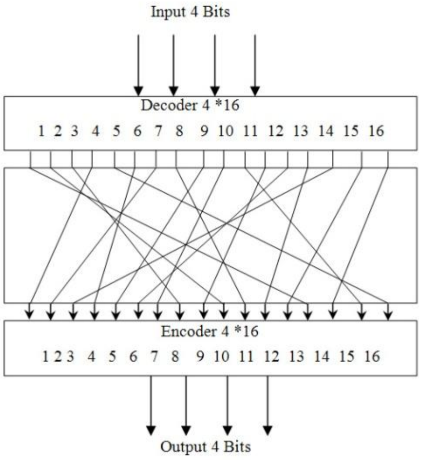


Figure.1. Simple substitution for 4 bits (Stalling, 2005)

Public key algorithms are known as asymmetric algorithms and private key algorithms are known as symmetric algorithms. Symmetric encryption algorithms are used mainly from Feistel method. These methods combine substitution and permutation simple method, then methods are combined

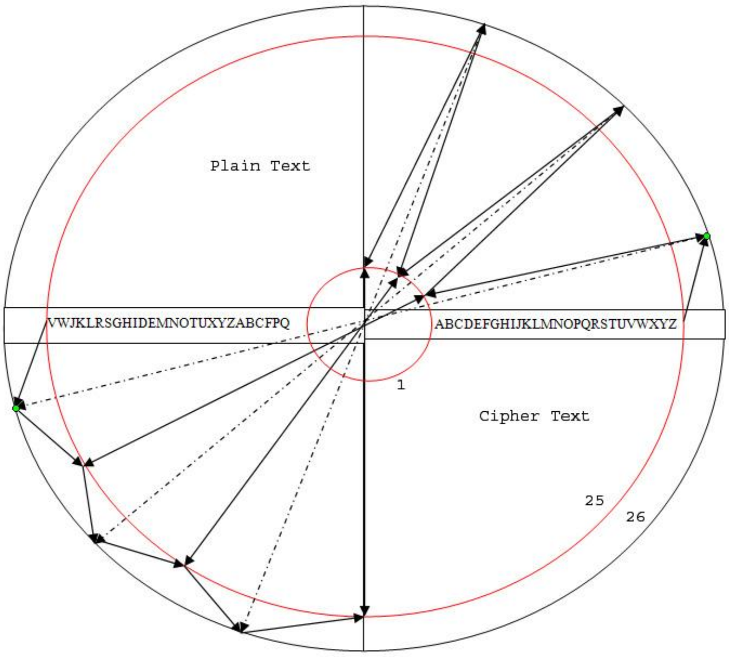


Figure.2. Proposed Algorithm

and will achieve complex and safe algorithm. The primitive principle of these algorithms is shown in figure 1.

These methods have two important specifications to eliminate attacks based on statistical analysis. Diffusion and confusion, Diffusion tries to diffuse plain text in all of cipher text as changing in one bit of plain text causes changing to a large extent of bits in cipher text. It is obtained from the combination of substitution and permutation bits in the plain text. Confusion tries to increase the the complexity between cipher text and plain text so that attackers do not achieve plain text with structural analysis of cipher text (Mandal, 2012).

Something which causes to use encryption algorithm in a wide range is the inner complexity of algorithm which does not easily break. According to mentioned phrases, Article tries to peresent an algorithm that has enough complexity and also it can be used easily.

The remainder of this paper is organized as follows. Section II presents the method used to encryption (EABC). Section III describes the sample used in our experiments. In Section IV, article presents algorithm codes. Concluding remarks are made in Section V.

I. Proposed Algorithm

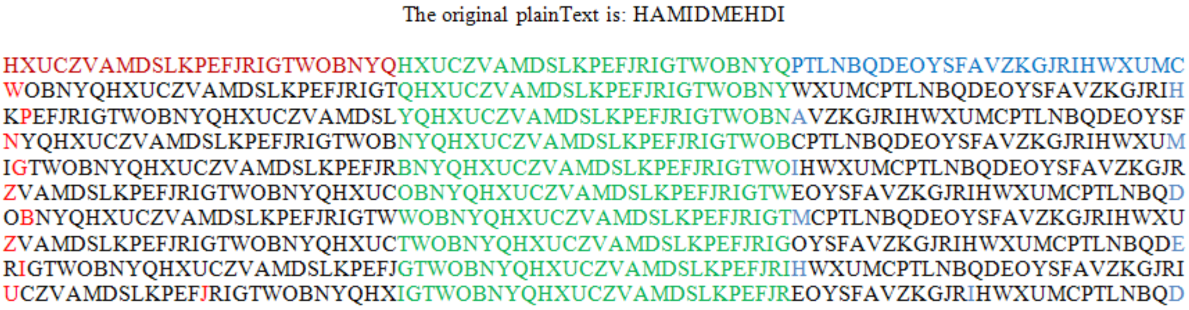
Proposed algorithm, which is called encryption algorithm based on circle (EABC), is shown in figure 2. This method is based on Feistel and has two parts, plain text and cipher text.

Algorithm is designed by a circle, top semicircle is Plain text and another semicircle is cipher text. Algorithm includes 26 circles for any characters in alphabet. Any top semicircles are for character in plain text and any bottom semicircles is a character in cipher text.

The algorithm has two Keyes. The first key to achieve plain text is called TKEY and a key for achieve cipher text is called CKEY. At the First stage, the first character must be indicated in plain

ISSN1925-931X

text via TKEY and must find its position then must find peer to peer position in cipher text. Now, the following characters in Tkey and Ckey must be permuted in terms of following conditions.
If position character in plain text is odd, permutation must be continued while reaching character to the end of TKEY but in even cases permutation continues while reaching character to the first position in TKEY and also in CKEY if the position of calculated character was odd in plain text, permutation must be continued while character reaches to the first in CKEY but in even cases permutation continues while character reaches the second position in CKEY. For any characters in plain text is generated a character in cipher text and also an expression is generated and adds for increasing the complexity and redundancy to all of cipher text in order to make algorithm difficult.



```
The original plainText is: HAMIDMEHDI

HXUCZVAMDSLKPEFJRIGTWOBNYQHXUCZVAMDSLKPEFJRIGTWOBNYQPTLNBQDEOYSFAVZKGJRIHWXUMC
WOBNYQHXUCZVAMDSLKPEFJRIGTQHXUCZVAMDSLKPEFJRIGTWOBNYWXUMCPTLNBQDEOYSFAVZKGJRIH
KPEFJRIGTWOBNYQHXUCZVAMDSLYQHXUCZVAMDSLKPEFJRIGTWOBNAVZKGJRIHWXUMCPTLNBQDEOYSF
NYQHXUCZVAMDSLKPEFJRIGTWOBNYQHXUCZVAMDSLKPEFJRIGTWOBCPTLNBQDEOYSFAVZKGJRIHWXUM
IGTWOBNYQHXUCZVAMDSLKPEFJRBNYQHXUCZVAMDSLKPEFJRIGTWOIHWXUMCPTLNBQDEOYSFAVZKGJR
ZVAMDSLKPEFJRIGTWOBNYQHXUCOBNYQHXUCZVAMDSLKPEFJRIGTWEOYSFAVZKGJRIHWXUMCPTLNBQD
OBNYQHXUCZVAMDSLKPEFJRIGTWWOBNYQHXUCZVAMDSLKPEFJRIGTMCPTLNBQDEOYSFAVZKGJRIHWXU
ZVAMDSLKPEFJRIGTWOBNYQHXUCTWOBNYQHXUCZVAMDSLKPEFJRIGOYSFAVZKGJRIHWXUMCPTLNBQDE
RIGTWOBNYQHXUCZVAMDSLKPEFJGTWOBNYQHXUCZVAMDSLKPEFJRIHWXUMCPTLNBQDEOYSFAVZKGJRI
UCZVAMDSLKPEFJRIGTWOBNYQHXIGTWOBNYQHXUCZVAMDSLKPEFJREOYSFAVZKGJRIHWXUMCPTLNBQD
```

Figure.3. Encryption Structure

Now, we illustrate the algorithm:

II. Illustrate the algorithm

Assume TKEY and CKEY are below values:

TKEY=PTLNBQDEOYSFAVZKGJRIHWXUMC
CKEY= HXUCZVAMDSLKPEFJRIGTWOBNYQ

And the plain text is HAMIDMEHDI. Now, must be achieved cipher text for it with proposed algorithm. Algorithm explains is as follows:

1- Finding the first character position of plain text in TKEY:
TKEY=PTLNBQDEOYSFAVZKGJRI**H**WXUMC

2- Finding peer to peer the first character position of plain text in CKEY:
CKEY= HXUCZVAMDSLKPEFJRIGT**W**OBNYQ

3- Permutation TKEY and CKEY while given character reaches to the first in TKEY and CKEY.
TKEY=**H**WXUMCPTLNBQDEOYSFAVZKGJRI
CKEY=**W**OBNYQHXUCZVAMDSLKPEFJRIGT

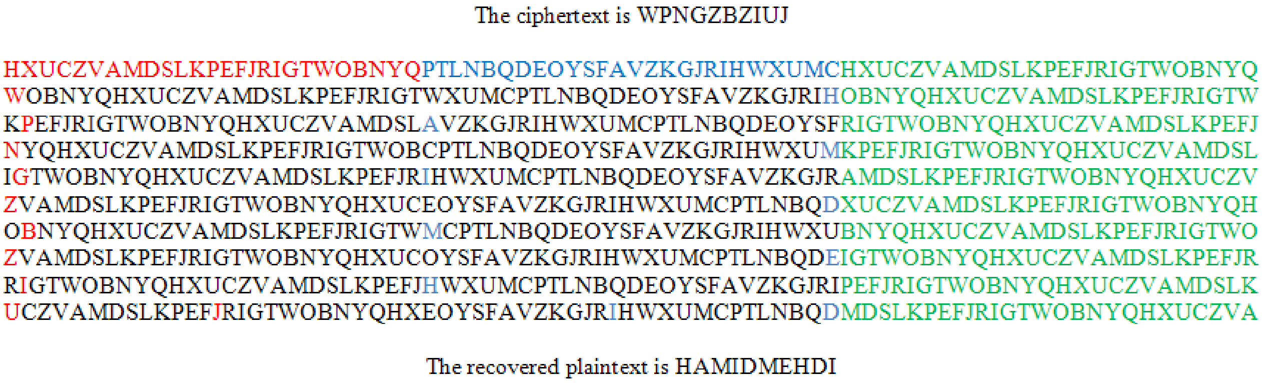
4- Now, if character position in plain text is odd, in TKEY must be permutation continued while character reaches the last position else in even case TKEY must be permutation continued while character reaches the first position and also if character position in plain text is odd in CKEY must be permutation while character reaches the fist position else CKEY must be permutation while character reaches the second position.

TKEY=**H**WXUMCPTLNBQDEOYSFAVZKGJRI
CKEY= T**W**OBNYQHXUCZVAMDSLKPEFJRIG

All of the steps are shown in figure 3. In figure 3 blue texts in the first row are TKEY and red texts in the first row are CKEY and all of green texts are extra text. From the second row to last from right which is defined with blue color, shows plain text and from the second row to last from left with red color is cipher text. Finally cipher text of HAMIDMEHDI is WPNGZBZIUJ. Therefore when there are TKEY and CKEY and cipher text, the plain text can be decoded and discovered.

III. Algorithm Codes

This algorithm includes two main functions encryption and decryption wich are designed by C#.Net.



```
The ciphertext is WPNGZBZIUJ

HXUCZVAMDSLKPEFJRIGTWOBNYQPTLNBQDEOYSFAVZKGJRIHWXUMCHXUCZVAMDSLKPEFJRIGTWOBNYQ
WOBNYQHXUCZVAMDSLKPEFJRIGTWXUMCPTLNBQDEOYSFAVZKGJRIHOBNYQHXUCZVAMDSLKPEFJRIGTW
KPEFJRIGTWOBNYQHXUCZVAMDSLAVZKGJRIHWXUMCPTLNBQDEOYSFRIGTWOBNYQHXUCZVAMDSLKPEFJ
NYQHXUCZVAMDSLKPEFJRIGTWOBCPTLNBQDEOYSFAVZKGJRIHWXUMKPEFJRIGTWOBNYQHXUCZVAMDSL
IGTWOBNYQHXUCZVAMDSLKPEFJRIHWXUMCPTLNBQDEOYSFAVZKGJRAMDSLKPEFJRIGTWOBNYQHXUCZV
ZVAMDSLKPEFJRIGTWOBNYQHXUCEOYSFAVZKGJRIHWXUMCPTLNBQDXUCZVAMDSLKPEFJRIGTWOBNYQH
OBNYQHXUCZVAMDSLKPEFJRIGTWMCPTLNBQDEOYSFAVZKGJRIHWXUBNYQHXUCZVAMDSLKPEFJRIGTWO
ZVAMDSLKPEFJRIGTWOBNYQHXUCOYSFAVZKGJRIHWXUMCPTLNBQDEIGTWOBNYQHXUCZVAMDSLKPEFJR
RIGTWOBNYQHXUCZVAMDSLKPEFJHWXUMCPTLNBQDEOYSFAVZKGJRIPEFJRIGTWOBNYQHXUCZVAMDSLK
UCZVAMDSLKPEFJRIGTWOBNYQHXEOYSFAVZKGJRIHWXUMCPTLNBQDMDSLKPEFJRIGTWOBNYQHXUCZVA

The recovered plaintext is HAMIDMEHDI
```

Figure.4. Decryption Structure

a. Encrypt Code

First function is encryption and it gets plain text as input and output of this function is cipher text. Encryption code is as follows:

```
private static string fAlphabet = MDSLKPEFJYQHXUCZVARIGTWOBN";
private static string sAlphabet = PYSHWTLNBQDEOXUMCFAVZKGJRI";
public static string Encrypt(string plainText){
char[] TKEY = fAlphabet.ToCharArray();
char[] CKEY = sAlphabet.ToCharArray();
char[] TKEY1 = fAlphabet.ToCharArray();
char[] plaintext= plainText.ToCharArray();
char[] ciphertext = new char[plaintext.Length];
char[] temp = new char[26];
char[] temp1 = new char[26];
int target;
for (int i = 0; i < plaintext.Length; i++){
Console.WriteLine("{0}  {1}  {2}", new string(TKEY), new string(TKEY1),
new string(CKEY));
target = Array.IndexOf (CKEY, plaintext[i]);
if (i % 2 == 0){
ciphertext[i] = TKEY[target];
}else{ciphertext[i] = TKEY[target + 1];
}// permute TKEY
for (int j = target; j < 26; j++) temp[j - target] = TKEY[j];
for (int j = 0; j < target; j++) temp[26 - target + j] = TKEY[j];
temp.CopyTo(TKEY, 0);
//--Extra data to increase complex
Random a = new Random();
int target1 = a.Next(26);
for (int j = target1; j < 26; j++) temp1[j - target1] = TKEY1[j];
for (int j = 0; j < target1; j++) temp1[26 - target1 + j] = TKEY1[j];
temp1.CopyTo(TKEY1, 0);
//---------permute CKEY
if (i % 2 == 0){target++;}
for (int j = target; j < 26; j++) temp[j - target] = CKEY[j];
for (int j = 0; j < target; j++) temp[26 - target + j] = CKEY[j];
temp.CopyTo(CKEY, 0);}
return new string(ciphertext);}
```

b. Decrypt Code

Second function is Decryption and it gets cipher text as input and output of this function is plain text. Decryption code is as follows:

```
public static string Decrypt(string cipherText){
char[]TKEY= fAlphabet.ToCharArray();
char[] TKEY1 = fAlphabet.ToCharArray();
char[] CKEY = sAlphabet.ToCharArray();
char[] ciphertext = cipherText.ToCharArray();
char[] plaintext = new char[ciphertext.Length];
char[] temp = new char[26];
char[] temp1 = new char[26];
int target;
for (int i = 0; i < ciphertext.Length; i++){
Console.WriteLine("{0}  {1}  {2}", new string(TKEY), new string(CKEY),
new string(TKEY1));
target = Array.IndexOf (TKEY, ciphertext[i]);
plaintext[i] = CKEY[target];
if (i % 2 == 0){target = target;} else{
target = target - 1;}                 plaintext[i] = CKEY[target];
// permute TKEY
for (int j = target; j < 26; j++) temp[j - target] = TKEY[j];
for (int j = 0; j < target; j++) temp[26 - target + j] = TKEY[j];
temp.CopyTo(TKEY, 0);
//----Extra data to increase complex
Random a = new Random();
int target1 = a.Next(26);
for (int j = target1; j < 26; j++) temp1[j - target1] = TKEY1[j];
for (int j = 0; j < target1; j++) temp1[26 - target1 + j] = TKEY1[j];
temp1.CopyTo(TKEY1, 0);
 // permute CKEY
 if (i % 2 == 0){target++; }
for (int j = target; j < 26; j++) temp[j - target] = CKEY[j];
for (int j = 0; j < target; j++) temp[26 - target + j] = CKEY[j];
temp.CopyTo(CKEY, 0);}
return new string(plaintext);}
```

The way using from functions is as follows:

```
string plainText = "HAMIDMEHDI";Console.WriteLine("The original
plainText is : {0}", plainText);
Console.WriteLine("\nThe TKEY and CKEY alphabets after each
permutation during encryption are\n");
string cipherText = Encrypt(plainText);
Console.WriteLine("\nThe ciphertext is {0}\n", cipherText);
string plainText2 = Decrypt(cipherText);
Console.WriteLine("\nThe    recovered    plaintext    is    {0}",
plainText2);
```

The most important advantage of EABC algorithm is changeability of methods for using in different forms. For example in many organization can be used while they cannot understand their algorithms because all of the functions in algorithm can be changed easily.

According to the expressed result, the most important advantage of this algorithm is changeablity with less cost and its complexity. One of the advantages is redundancy in cipher text and paying attention to this point just cipher text is sent therefore the decryption of cipher is not easy. Also one another important advantage of this algorithm is the Keyes changeability, that is possible easily, therefore this point causes complexity to increase.

IV. Conclusion

Nowadays, we live in the world that information faces to different dangers, robberies, misuses, etc. One of the most important methods to prevention these dangers is cryptography. This is important that any organizations must have a cryptography algorithm for themselves so that they become sure from information when transferring.

In this paper it has been tried to present an encryption algorithm which is user-friendly and has difficult encryption. This algorithm is quick in changing and redundancy and has wide range of permutation, that all of them help the complexity and finally increasing its safety.

Acknowledgement

This research was supported by the Islamic Azad University, Andimeshk Branch.

References

Shashi Mehrotra Seth, Rajan Mishra on " Comparative Analysis Of Encryption Algorithms For Data communication " in IJCST Vol. 2, Issue 2, June 2011 I ,pp. 292-294

William Stalling, Cryptography and network Security, 4th Edition, Prentice –Hall, 2005

Jawahar Thakur, Nagesh Kumar, “DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis, “in International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459, Volume 1, Issue 2, December 2011), pp.6-12

Shanta, yoti Vashishtha on, “Evaluating the performance of Symmetric Key Algorithms: AES (Advanced Encryption Standard) and DES (Data Encryption Standard) in IJCEM International Journal of Computational Engineering & Management, Vol. 15 Issue 4, July 2012 ,pp.43-49

Himani Agrawal and Monisha Sharma “Implementation and analysis various symmetric Cryptosystems “in Indian Journal of Science and Technology in Vol. 3 No.12 (Dec 2010) ISSN: 0974- 846, p.1173-1176

S.Pavithra, Mrs. E. Ramadevi “STUDY AND PERFORMANCE ANALYSIS OF CRYPTOGRAPHY ALGORITHMS ” International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 5, July 2012 14, pp.82-86

Pratap Chnadra Mandal, "Superiority of Blowfish Algorithm", International Journal of Advenced Research in Computer Science and Software Engineering 2(9), Volume 2, Issue 9, September 2012, pp. 196-201