Romantic Tantalizers Cipher as an Improved Version of Hill Cipher

Andru Putra Twinanda

Informatics Engineering, School of Electrical Engineering and Informatics, Institut Teknologi Bandung
e-mail: ndrewh@yahoo.com

Abstract *– Hill cipher is a polygraphic substitution cipher based on linear algebra. Invented by Lester S. Hill in 1929, it was the first polygraphic cipher in which it was practical (though barely) to operate on more than three symbols at once.*

Hill Cipher needs many computations because it has to find the inverse of the matrix that has been decided to be the key. The size of this matrix is the square of the length of the string wanted to be encrypted. In order to minimize the computation, Romantic Tantalizer is introduced. In Romantic Tantalizers, the size of the matrix will be reduced. It will be the closest quadratic number higher than the plaintext's length.

Key Words*: Hill Cipher, linear algebra, Romantic Tantalizers, matrix.*

1. INTRODUCTION

Security of information has become an issue since privacy has been introduced to humankind. In order to manage the security of information, people have tried to hide the true meaning of information and this field of study is called cryptography. People like Ron Rivest, Justice Peter Smith, and Giovan Batista Bellaso have found so many methods in cryptography. Inspired by those people, Author would like to implement one possible way to encrypt information, in this context, is a text. This method is called Romantic Tantalizers Cipher. It is the anagram of "Matrices Tranzlation". This algorithm may be found quite similar to Hill Cipher, but of course with improvements.

In Hill Cipher, a plaintext is translated into a vector and will be multiplied by a key-matrix with the same size of this vector. In Romantic Tantalizers Cipher, instead of translating the plaintext into a vector, it will be translated into a matrix. This method will help reducing space requirement because if the text is translated into a vector, like what Hill Cipher does, then the key-matrix's size would be the square of the length of the plaintext. In Romantic Tantalizers Cipher, the key-matrix's size would be the closest quadratic number higher than the plaintext's length. In Romantic Tantalizers Cipher, the plaintext would also be split into blocks and each block will contain one word to encrypt.

2. HILL CIPHER

2.1. Basic Concept

Hill Cipher will translate the plaintext and the key into a vector and a matrix, respectively. Each letter is first encoded as a number. Often the simplest scheme is used: A = 0, B =1, ..., Z=25, but this is not an essential feature of the cipher. The matrix has got to be invertible to ensure the decryption process is invertible.

2.2. Implementation

2.2.1. Algorithm

Here is the algorithm of the Hill Cipher:

Figure 1 Hill Cipher Algorithm

```
function HillCipherEnc (plaintext:string,
key:string) : string

var
  Vect1        : Vector;
  keyM  : Matrix;

begin
  Vect1 := ConvertToVector(plaintext);
{convert string to vector}

  keyM := ConvertToMatrix(key);
{convert string to matrix}

  Vect1 := keyM * Vect1;
  HillCipherEnc :=ConvertToString(Vect1);
{convert vector to string}
End
```

```
function HillCipherDec (cipher:string,
key:string) : string
```

```
var
  Vect1        : Vector;
  keyM  : Matrix;

begin
  Vect1 := ConvertToVector(cipher);
{convert string to vector}

  keyM := ConvertToMatrix(key);
{convert string to matrix}

  keyM := Inverse(keyM);
{compute invers from matrix keyM}

  Vect1 := keyM * Vect1;
  HillCipherDec :=ConvertToString(Vect1);
{convert vector to string}
End
```

2.2.2. Encryption

Assume that we’re going to encipher the text ‘ACT’. Since A is ‘0’, C is ‘2’, and T is ‘19’the string will be translated into the vector below:

$$\begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix}$$

Figure 2 Vector of String Translation

Considering the key is ‘GYBNQKURP’ (it is chosen so that the matrix will be invertible), the key matrix will be:

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix}$$

Figure 3 Key Matrix

The enciphered vector will be:

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix} = \begin{pmatrix} 67 \\ 222 \\ 319 \end{pmatrix} \equiv \begin{pmatrix} 15 \\ 14 \\ 7 \end{pmatrix} \pmod{26}$$

Figure 4 Encryption Multiplication Process

which corresponds to a ciphertext of ‘POH’.

2.2.3. Decryption

As explained before, the decryption will be done by using the inverse of the matrix that’s used in encryption process. We find that in Z_{26} the inverse matrix of the one in the previous example is:

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix} \longrightarrow \begin{pmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{pmatrix}$$

Figure 5 Inverse Matrix

The inverse of the key matrix is ‘IFKVIVVMI’ in letters. (There are standard methods to calculate the inverse matrix, it will not be explained in this paper)

Just as before, the next process is to multiply the matrix with the vector, in this case the ciphertext vector.

$$\begin{pmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{pmatrix} \begin{pmatrix} 15 \\ 14 \\ 7 \end{pmatrix} \equiv \begin{pmatrix} 260 \\ 574 \\ 539 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix} \pmod{26}$$

Figure 6 Decryption Multiplication Process

This multiplication results is the same vector as the plaintext vector. This concludes the encryption and decryption process using Hill Cipher.

2.3. Strength

The strength of Hill Cipher is it provides diffusion. For example, an appropriately chosen matrix can guarantee that small differences before the matrix multiplication will result in large differences after the matrix multiplication. Some modern ciphers use indeed a matrix multiplication step to provide diffusion. For example, the MixColumns step in AES is a matrix multiplication. The function *g* in Twofish is a combination of non-linear S-boxes with a carefully chosen matrix multiplication (MDS).

2.4. Weakness

Hill Cipher is vulnerable to a known-plaintext attack because it is completely linear. An opponent who intercepts n^2 plaintext/ciphertext character pairs can set up a linear system which can (usually) be easily solved; if it happens that this system is indeterminate, it is only necessary to add a few more plaintext/ciphertext pairs. Calculating this solution by standard linear algebra algorithms then takes very little time.

In addition to its vulnerability to the attack, the key matrix should be chosen carefully because not every matrix is invertible modulo 26. It is if and only if it is invertible both modulo 2 and modulo 13. This constraint can make the process more laborious.

Not only choosing the right matrix is hard, the computation will also be hard. For a three-letters word, a 3x3 matrix is needed. For a six-letters word, then a 6x6 matrix is needed. As the dimension increases, the cipher rapidly becomes infeasible for a human to operate by hand.

3. ROMANTIC TANTALIZERS

3.1. Basic Concept

The basic concept of this method is not very much different with Hill Cipher. But, instead of translating the text into a vector, the text will be translated into a matrix. This change will become helpful in reducing the space and the computational needs.

This algorithm will also separate the string into blocks of 16 characters when the string is more than 16 characters length. This process is done so that we won’t need a really big matrix in order to encrypt the text. Instead of a big matrix, we will only need a matrix with 4x4 size.

Besides that, we are also going to change the translation mechanism. Hill Cipher is translating A = ‘0’, B = ‘1’ ..., Z = ‘25’. Instead of that, Romantic Tantalizers will use the ASCII code. The reason of choosing the substitution with ASCII code will be explained later.

3.2. Implementation

3.2.1. Algorithm

Here is the algorithm of the Romantic Tantalizers:

Figure 7 Romantic Tantalizer Algorithm

```
function RomanticTantalizerEnc
(plaintext:string, key:string) : string

var
  keyM       : Matrix;
  plainM    : Matrix;
  listS      : List of String;
  listOutp  : List of String;
  outp       : string;

begin
  listS := ConvertToList(plaintext);
{membagi plaintext to blok berisi 16
karakter}

  keyM := ConvertToMatrix(key);

{convert string key to matrix}
  while (not empty(listS))
    begin
       outp:=First(listS);
       DelFirst(listS);
       plainM := ConvertToMatrix(outp);
{convert string outp to matrix}

       plainM := keyM*plainM;
       outp := ConvertToString(plainM);
{convert matrix to string}

       Add(listOutp,outp); {menambahkan
outp ke dalam listOutp}
    end;

 RomanticTantalizerEnc:=Concat(listOutp);
{menyatukan semua elemen dalam list}
end
```

```
function RomanticTantalizerDec
(cipher:string, key:string) : string

var
  keyM          : Matrix;
  cipherM      : Matrix;
  listS         : List of String;
  listOutp     : List of String;
  outp          : string;

begin
  listS := ConvertToList(cipher);
{membagi cipher to blok berisi 16
karakter}

  keyM := ConvertToMatrix(key);
{convert string key to matrix}

  keyM := Inverse(keyM);
{membentuk inverse matrix from keyM}

  while (not empty(listS))
  begin
       outp:=First(listS);
       DelFirst(listS);
       cipherM := ConvertToMatrix(outp);
{convert string outp to matrix}

       cipherM := keyM*cipherM;
       outp := ConvertToString(cipherM);
{convert matrix to string}

       Add(listOutp,outp); {menambahkan
outp ke dalam listOutp}
       end;

 RomanticTantalizerDec:=Concat(listOutp);
{menyatukan semua elemen dalam list}
End
```

3.2.2. Encryption

Assume that we're going to encipher the text 'ACTUAL'. Using the ASCII table code, the string will be translated into the matrix below:

$$\begin{bmatrix} 65 & 67 & 81 \\ 82 & 65 & 77 \\ 0 & 0 & 0 \end{bmatrix}$$

Figure 8 Matrix of The Plaintext

As shown above, padding is needed in this algorithm because the size of the string is not always a quadratic number. By adding zero to the matrix, we will be able to get an NxN matrix from a string, length of which is not a quadratic number.

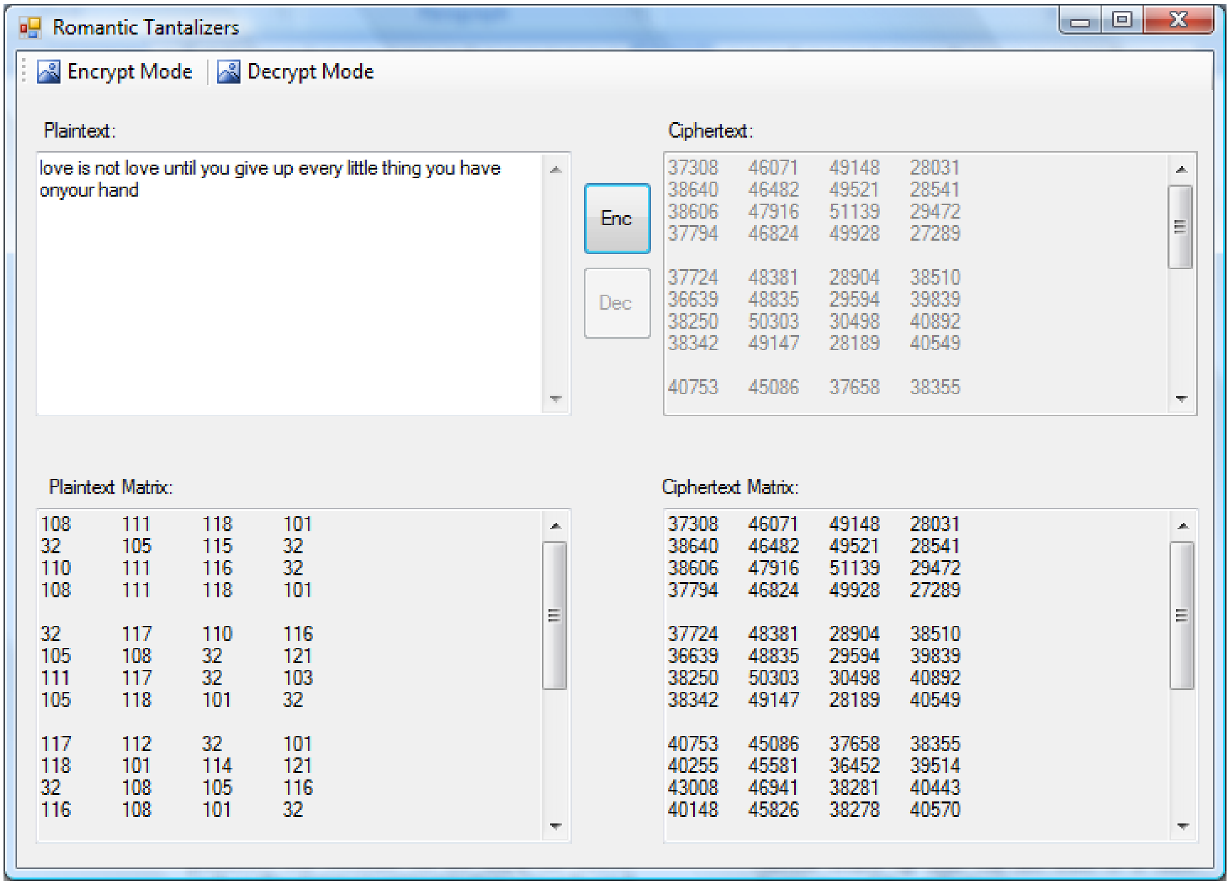
This is the reason of substituting the characters with their ASCII code, so that we would be able to do padding in the matrix. If we are to use the previous configuration, it would be hard to do the padding because number 0 is already used to represent character 'a'. But in ASCII code, number 0 represents a null character, so this number can be used to do the padding.

Consider the key matrix is the same as shown in **Figure 3**. The ciphertext matrix will be:

$$\begin{bmatrix} 2358 & 1962 & 2334 \\ 2157 & 1911 & 2285 \\ 2694 & 2445 & 2929 \end{bmatrix}$$

Figure 9 Encryption Multiplication Result

The encryption process that's done by the software made by the author, RomanTan, is as shown below:



Plaintext:

```
love is not love until you give up every little thing you have
onyour hand
```

Ciphertext:

```
37308   46071   49148   28031
38640   46482   49521   28541
38606   47916   51139   29472
37794   46824   49928   27289

37724   48381   28904   38510
36639   48835   29594   39839
38250   50303   30498   40892
38342   49147   28189   40549

40753   45086   37658   38355
```

Plaintext Matrix:

```
108   111   118   101
32    105   115   32
110   111   116   32
108   111   118   101

32    117   110   116
105   108   32    121
111   117   32    103
105   118   101   32

117   112   32    101
118   101   114   121
32    108   105   116
116   108   101   32
```

Ciphertext Matrix:

```
37308   46071   49148   28031
38640   46482   49521   28541
38606   47916   51139   29472
37794   46824   49928   27289

37724   48381   28904   38510
36639   48835   29594   39839
38250   50303   30498   40892
38342   49147   28189   40549

40753   45086   37658   38355
40255   45581   36452   39514
43008   46941   38281   40443
40148   45826   38278   40570
```

Figure 10 RomanTan in Encryption Mode
Plaintext: "love is not love until you give up every little thing you have onyour hand", Key: "andru"

3.2.3. Decryption

Just like the Hill Cipher, Romantic Tantalizers will use the inverse of the key matrix to be multiplied with the ciphertext in order to get the plaintext, but because we are using the ASCII code as the plaintext representation, upper bound of which is 127, the key matrix doesn't have to be modulo 26 invertible, so we can't use the inverse matrix that is shown in **Figure 5**.

The matrix that will be used in the decryption process is:

$$\begin{bmatrix} 0.1587 & -0.777 & 0.5079 \\ 0.0113 & 0.1587 & -0.1065 \\ -0.2245 & 0.8571 & -0.4898 \end{bmatrix}$$

Figure 11 Inverse Matrix

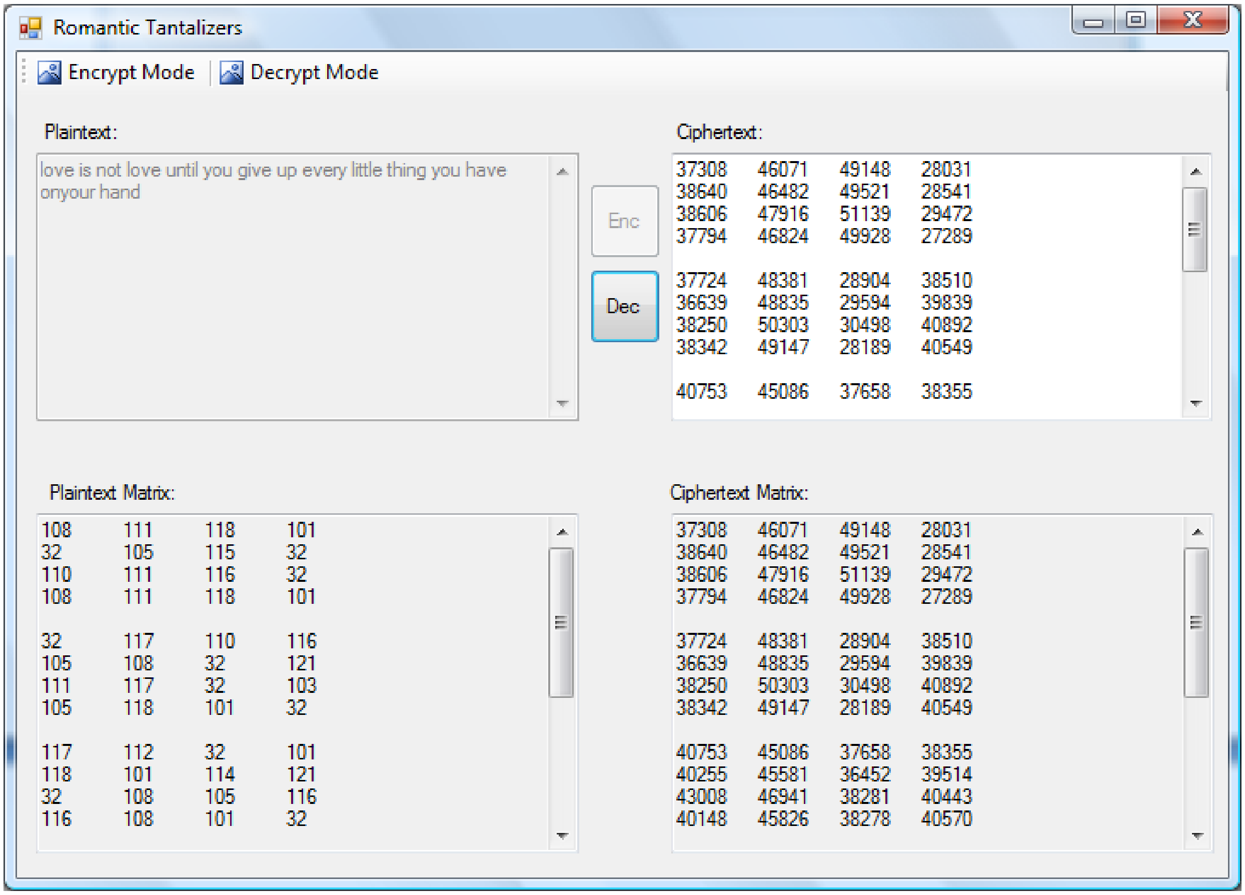
Then the result of the multiplication is:

$$\begin{bmatrix} 65 & 67 & 81 \\ 82 & 65 & 77 \\ 0 & 0 & 0 \end{bmatrix}$$

Figure 12 Result of the Decryption Process

Just as shown, the multiplication result is the plaintext matrix.

The decryption process that's done by the RomanTan, is as shown below:



Plaintext:

love is not love until you give up every little thing you have onyour hand

Ciphertext:

```
37308   46071   49148   28031
38640   46482   49521   28541
38606   47916   51139   29472
37794   46824   49928   27289

37724   48381   28904   38510
36639   48835   29594   39839
38250   50303   30498   40892
38342   49147   28189   40549

40753   45086   37658   38355
```

Plaintext Matrix:

```
108     111     118     101
32      105     115     32
110     111     116     32
108     111     118     101

32      117     110     116
105     108     32      121
111     117     32      103
105     118     101     32

117     112     32      101
118     101     114     121
32      108     105     116
116     108     101     32
```

Ciphertext Matrix:

```
37308   46071   49148   28031
38640   46482   49521   28541
38606   47916   51139   29472
37794   46824   49928   27289

37724   48381   28904   38510
36639   48835   29594   39839
38250   50303   30498   40892
38342   49147   28189   40549

40753   45086   37658   38355
40255   45581   36452   39514
43008   46941   38281   40443
40148   45826   38278   40570
```

Figure 13 RomanTan in Decryption Mode With Ciphertext and Key From The Previous Display

This concludes the encryption and decryption process using Romantic Tantalizers.

3.3. Strength

Romantic Tantalizers still preserve the strength of the Hill Cipher, but it doesn't really give any other strength that can take account for because technically it doesn't give many changes to the Hill Cipher.

3.4. Weakness

The computation for Romantic Tantalizers is still laborious once the plaintext gets really long. And it still is vulnerable to known-plaintext attack since it is still completely linear.

The trade off of not using the matrix that is modulo 26 invertible is the inverse matrix will be in real value, not integer value.

Another weakness of Romantic Tantalizer is the involvement of rounding up. This weakness appears in the decryption process because of the inverse matrix will appear in real value that will cause the multiplication result is in real value too. But this weakness can be handled with a careful round up that will not lead to a false decryption of the ciphertext.

4. COMPARISON

Looking at the strength, both algorithms don't really have many differences, in fact the improvement that Romantic Tantalizer has doesn't impact on the strength of Hill Cipher.

On the other hand, looking at the weaknesses, the two of them are quite distinct. As explained, the key matrix in Hill Cipher has more constraints that have got to be satisfied than the Romantic Tantalizer has. This means that choosing a key matrix in Romantic Tantalizer will be easier than in Hill Cipher.

The Hill Cipher is better in finding the inverse matrix because the inverse matrix found is in integer value, not in real value, so that rounding up won't happen in Hill Cipher. This is a trade off of the previous point.

The most significant difference between the two of them is the using of matrix instead of vector to represent the ciphertext. This will reduce the space use up to the closest quadratic number higher than the plaintext's length. If the matrix is smaller than the computation of the inverse matrix will be less too.

5. CRYPTANALYSIS ON ROMANTIC TANTALIZERS

5.1. Ciphertext-only Attack

There will be repetitions in Romantic Tantalizers because there might be repetition after 16 characters in the plaintext. So Romantic Tantalizers is vulnerable to frequency analysis. But after finding the repetition, the attacker would probably end up with no idea of how the algorithm works because it's not a direct substitution algorithm, like the Vigenere Cipher. Because of the involvement of other characters that ends up in the same column of the plaintext matrix and the involvement of other characters that ends up in the same row of the key, this algorithm is very strong against ciphertext-only attack.

5.2. Known-plaintext Attack

Just like the Hill Cipher, Romantic Tantalizers is also vulnerable to known-plantext attack as explained in sub chapter 2.4.

5.3. Chosen-plaintext Attack

As this algorithm is vulnerable to a known-plaintext attack, this algorithm will also be vulnerable to chosen-plaintext attack. It would even be easier if the cryptanalysts knows that the biggest size of the matrix used in this algorithm is 4x4. What they have to do is just taking sample on the first 16 characters in the plaintext and 16 first characters in the ciphertext. If they can find a way to map from the plaintext to the ciphertext, the algorithm will be broken.

6. CONCLUSION AND SUGGESTION

6.1. Conclusion

According to explanation and experiments from the previous chapters, author can conclude that:

a. Romantic Tantalizer doesn't solve all the weaknesses that Hill Cipher has
b. Romantic Tantalizer is more space-friendly than the Hill Cipher because of the using of Matrix Translation

c. Both algorithms are still vulnerable to known plaintext
d. Because of the involvement of matrix and the calculation of the inverse matrix, both algorithms can be laborious for a long plaintext.

6.2. Suggestion

Suggestions for the ones interested in developing Romantic Tantalizer algorithm that author has are:

a. For the ones interested in developing the algorithm, find a matrix inversion method capable of generating a 128 modulo invertible matrix in order to get an integer valued matrix, instead of a real valued matrix.
b. Find a way to make an invertible matrix from whatever key the user gives because the current method simply doesn’t facilitate that.

7. BIBLIOGRAPHY

[1] Munir, Rinaldi, Diktat Kuliah IF5054 Kriptografi, Penerbit ITB 2006

[2] Hill Cipher, http://en.wikipedia.org/wiki/Hill_cipher