FOURIER MASKING ENCRYPTION ALGORITHM FOR POLYALPHABETIC SYMMETRIC KEY CRYPTOGRAPHY

Nadim Jahangir and Ahsan Raja Chowdhury
Department of Computer Science and Engineering
Northern University Bangladesh
E-mail: cse_nj@yahoo.com, farhan717@yahoo.com

Abstract: *Cryptography is the art of keeping message secret by different methods. In this paper, a new algorithm has been proposed for symmetric key polyalphabetic data encryption, namely Fourier Masking Encryption Algorithm (FMEA). The concept of Fourier series is exploited efficiently to generate the random key sequence from a given password, as randomness is the key property of polyalphabetic encryption. It has been shown that the proposed algorithm can provide high level cryptanalytic complexity for any unauthorized attempt to decryption. Experimental results are also provided here, which shows the efficiency of the proposed one over the existing algorithms.*

Keywords: *Fourier series, Polyalphabetic Encryption, Symmetric Key, Masking Function*.

1. Introduction

There are several encryption algorithms that are the outcome of extensive research in the recent years. Some of them indeed provide good security but some others are vulnerable to either brute-force or cryptanalytic attack. Some of them are easy to implement in hardware with high processing power and storage capacity and some others are good for limited devices like mobile phones, PDAs etc. Most of the available cryptographic algorithms are based upon number theory which use finite-field such as **GF(*p*)** or **GF(*2^n*)**. Most of these number theoretic algorithms are more secure when they use large prime numbers or large binary words [1, 2]. But when the precision and bit-width increases, the hardware in which the algorithm is to be implemented must be sophisticated in processing power and storage capacity and hence it tends to high cost. We have developed such an encryption/decryption algorithm which exploits the versatility of Fourier series [3, 4]. It doesn't need any additional storage; rather provide a very good security. This can be useful in almost every system with memory and processing power constraints. It is a symmetric key polyalphabetic encryption technique where a single key is shared for both encryption and decryption algorithm. The paper is organized as follows: Some preliminary concepts are discussed in Section 2. Proposed method is discussed in Section 3 with Algorithm and example. Section 4 shows the Cryptanalytic Complexity of the Proposed Algorithm from different points of view. Proposed Algorithm is compared with the existing algorithms in Section 5. Section 6 concludes the paper.

2. Preliminaries

In this section some basic definitions are given that are used throughout this paper. Some of the existing encryption algorithms are also discussed in this section.

Definition 2.1. Plain Text and Cipher Text: 'Plain Text' is the message that is to be encrypted and 'Cipher Text' is the encrypted message.

Definition 2.2. Polyalphabetic Encryption: 'Polyalphabetic Encryption' technique uses different keys for the encryption of individual characters in the plain text. So in this scheme, several key values are required [1].

Definition 2.3. Fourier Series (FS): A series proposed by the French mathematician Fourier about the year 1807. The series involves the *sine*s and *cosine*s of whole multiples of a varying angle and is usually written in the form:

$$y = H_0 + A_1 \sin x + A_2 \sin 2x + A_3 \sin 3x + \ldots B_1 \cos x + B_2 \cos 2x + B_3 \cos 3x + \ldots. \qquad (1)$$

By taking a sufficient number of terms the series may be made to represent any periodic function of x [4]. The original definition of FS restricts the angles to be harmonically related. But we extend the original definition of FS according to our need for the compatibility with our proposed algorithm. The form of the FS that we developed is presented in the next section.

There are several encryption algorithms for symmetric key encryption. There are also block cipher algorithms that operates on a single block at a time. But block ciphers can also be implemented for stream encryption in several modes [1]. For example, DES (Data Encryption Standard) is a block cipher technique but can be used for stream encryption in ECB (Electronic Code-Book), CFB (Cipher Feed Back), Counter modes. Other encryption algorithms are AES (Advanced Encryption Standard), BlowFish, RC4 [1] etc. All these algorithms are based upon number theory (finite-field is used) and use some additional storage like S-box (Substitution Box), Permutation Table, Initial Value etc. These algorithms are also complex in nature, as the encryption process is very much complex. But when complexity is increased, the difficulties of hardware and software implementation are also raised. Computational complexity does not matter if the algorithm can provide with a strong defense against fraud. But if we consider developing any encryption algorithm for limited devices like mobile phones, PDAs, BlueTooth hardware etc, storage capacity and processing power of that device is also a matter of concern. In that case, we must explore for an algorithm which is simpler in hardware implementation but holds a very strong position against intruders.

3. Proposed Algorithm

We have developed a function which is like Fourier series [3, 4] or trigonometric polynomial that is used to generate unpredictable key sequence. The function takes a password of any length chosen by the encryption party and results in key sequence. We found that by evaluating Fourier series, a good randomness can be achieved. The sequence is then XORed with the plaintext to get the cipher text. The function is the heart of this algorithm and named as 'Masking Function'. This is presented in (2).

$$f(n,A)=\left\lfloor A\frac{\sum\limits_{i=1,3,5...}^{M-1} P_i\cos\left(\frac{\pi(P_{i+1}+\pi)n}{500}\right)}{\sum\limits_{i=1,3,5...}^{M-1} P_i}\right\rfloor \qquad (2)$$

where,
P = password, n = 1, 2, 3…length[plain-text],
M = length[P] and
A = a sequence of plaintext

Algorithm 1: Fourier_Masking_Encryption _Algorithm (FMEA)

Part – 1 (Construction of Masking Function)

I. Choose any password in the form of ASCII text.
II. Convert the values of the characters in the chosen password to their equivalent extended ASCII values and put the values in the vector $P = \{P_1, P_2, P_3 \dots P_M\}$. *[Here M = number of characters in the password]*
III. If M is odd then pad a one '1' at the end of the vector P. *[So that after this padding operation P_M becomes equal to '1' and the value of M is increased by one]*
IV. Using the definition of the (2) construct the masking function.

As the masking function is ready, encryption/decryption procedure is followed.

Part – 2 (Encryption/Decryption)

{Encryption}

I. Take the plain text sequence in the form of ASCII text.
II. Convert the characters in the plain text to their equivalent extended ASCII values and put the values in the vector $X = \{X_1, X_2, X_3 \dots X_N\}$. *[Here N = number of characters in the plain text]*
III. Generate the cipher text sequence $Y = \{Y_1, Y_2, Y_3 \dots Y_N\}$ by $Y_i = X_i \oplus f(i, X_{i-1})$ where $X_0 = P_1$.

{Decryption}

I. Take the cipher text sequence in the form of ASCII text.

II. Convert the characters in the cipher text to their equivalent extended ASCII values and put the values in the vector $Y = \{Y_1, Y_2, Y_3 \ldots Y_N\}$. *[Here N = number of characters in the cipher text]*

III. Retrieve the plain text sequence $X = \{X_1, X_2, X_3 \ldots X_N\}$ by $X_i = Y_i \oplus f(i, X_{i-1})$ where $X_0 = P_1$.

Example 1: The algorithm is illustrated by an example which is summarized in Table 1. In this example the plaintext and the password are X = STAY = {83, 84, 65, 89} and P = COMPLEX = {67, 79, 77, 80, 76, 69, 88} respectively. To make the length of P even we do P = P || 1 = {67, 79, 77, 80, 76, 69, 88, 1}. The masking function becomes, f(n,A) = FLOOR(ABS((A*(67cos0.516n + 77cos0.522n + 76cos0.453n + 88cos0.026n))/308))

Table 1. Example of FMEA encryption

X	*A*	*N*	***k* = *f* (*n*, *A*)**	***Y* = *k* ⊕ *X***	*Y* in ASCII Character
83	67	1	61	110	*n*
84	83	2	56	108	*l*
65	84	3	28	93	*]*
89	65	4	0	89	*Y*

4. Cryptanalytic Complexity Of The Proposed Algorithm

In this section, the cryptanalytic complexity proposed algorithm will be described from two points of view.

4.1. One-to-many Mapping Between Plaintext and Cipher text

The proposed algorithm maps the plaintext characters to a large space of cipher text characters. This feature thwarts any cryptanalyst in exploiting the statistical relation between plaintext and cipher text. For example, the plaintext "AAAAAAAAAAAAAAAAAAAAAAAA" contains 24 A's and when it is encrypted by FMEA with the password 2dÈú then the cipher text becomes Fq@iS|ExCgRrU□ChQi^}OqL`. Here we can see that the cipher text contains 23 different characters for a single plaintext character 'A'. In Section 5, this property will be highlighted more sophisticatedly by comparing the proposed algorithm with Vigenere cipher [1, 8].

4.2. Complicated Solution Procedure for Cracking Password

By observing the proposed algorithm and the Masking Function we can infer that each cipher text character is a function of two plaintext characters and the password. The plaintext and cipher text characters can be equated as follows:

$$
\begin{aligned}
&Y_1 = Function(X_1, P_1, P_1, P_2, P_3 \ldots P_M)\\
&Y_2 = Function(X_2, X_1, P_1, P_2, P_3 \ldots P_M\\
&Y_3 = Function(X_3, X_2, P_1, P_2, P_3 \ldots P_M) \qquad (3)\\
&\ldots \quad \ldots\ldots\ldots\\
&\ldots \quad \ldots\ldots\ldots\\
&Y_N = \mathbf{Function}(X_N, X_{N-1}, P_1, P_2, P_3 \ldots P_M)
\end{aligned}
$$

If Y and X are known (that is, if the attack **is known plaintext attack** [1]), then M equations are required to solve for P. The above system of equation is a non-linear complicated equation with trigonometric polynomial. The password P is unknown and hence M is also unknown since it is the length of the password and hence solving the system is impractical. However, if the value of M is known then there is also a difficulty in finding P if N<M. If M is known and N≥M then there is another complication, that is; the system can not be solved if there are no sufficient plaintext and cipher text characters. One system of such equations is shown in (4).

Here, we assume the password to be $P_1P_2P_3P_4$.

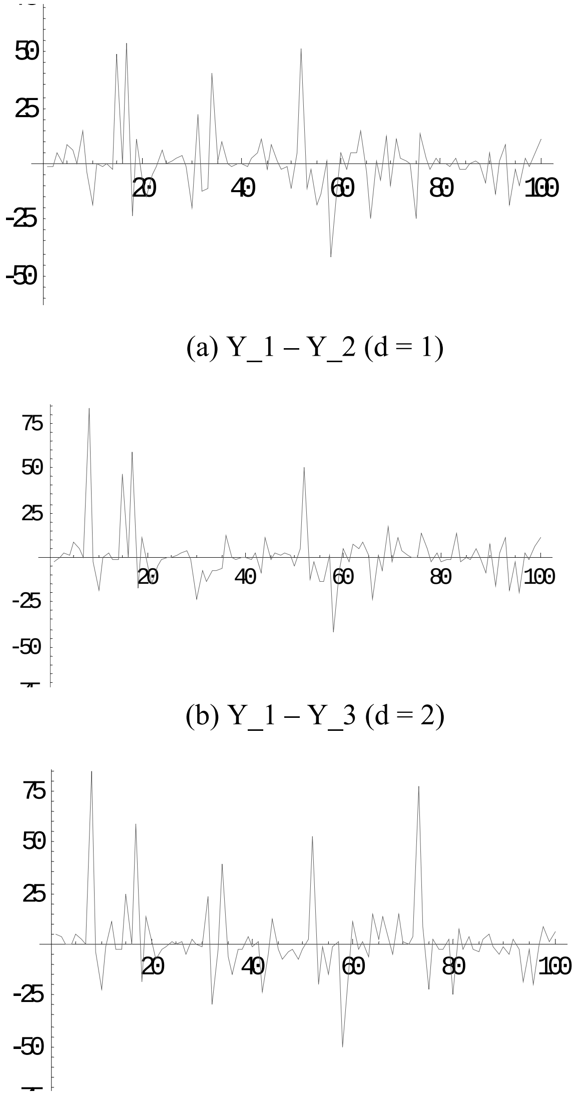
$$Y_1 = X_1 \oplus \left\lfloor \left| P_1 \frac{P_1 \cos(0.0063P_2 + 0.019) + P_3 \cos(0.0063P_4 + 0.019)}{P_1 + P_2 + P_3 + P_4} \right| \right\rfloor$$

$$Y_2 = X_2 \oplus \left\lfloor \left| X_1 \frac{P_1 \cos(0.0125P_2 + 0.039) + P_3 \cos(0.0125P_4 + 0.039)}{P_1 + P_2 + P_3 + P_4} \right| \right\rfloor \qquad (4)$$

$$Y_3 = X_3 \oplus \left\lfloor \left| X_2 \frac{P_1 \cos(0.0188P_2 + 0.059) + P_3 \cos(0.0188P_4 + 0.059)}{P_1 + P_2 + P_3 + P_4} \right| \right\rfloor$$

$$Y_4 = X_4 \oplus \left\lfloor \left| X_3 \frac{P_1 \cos(0.0251P_2 + 0.079) + P_3 \cos(0.0251P_4 + 0.079)}{P_1 + P_2 + P_3 + P_4} \right| \right\rfloor$$

This system holds a massive difficulty in finding P. There is no linear relation among plaintext, cipher text and the password. A fixed plaintext of length 100 ASCII characters is encrypted by FMEA for different passwords. All the passwords P_n are of length 16. Y_n is the cipher text for password P_n. Number of different elements in P_n than P_1 is denoted by'd'. The graphs of Fig. 1 show the differences in the sequences of cipher text Y_n and Y_1. We presented the change in cipher text for the change in the password.



(a) Y_1 – Y_2 (d = 1)

(b) Y_1 – Y_3 (d = 2)

(c) Y_1 – Y_4 (d = 3)

Figure 1. Graph of Y_1 – Y_n with varying 'd'

There is no way to apply FFT (Fast Fourier Transform) [5, 6], DFT (Discrete Fourier Transform) [5, 6, 7], DCT (Discrete Cosine Transform) or other transform techniques to crack the cipher as the cipher is a highly distorted Fourier series. The nature of FMEA yields this distortion.

5. Experimental Results

The Randomness in the Round Key sequence obtained by evaluating the function (2) has been analyzed extensively for different plaintext and passwords. Some simulated results are presented in Table 2. The randomness in round key is easily understood by observing Table 2. Analyzing the integrity of the existing techniques [3, 6], it is found that the proposed method provides higher level security in the message than the others because of the nature of the algorithm and the use of Fourier series. The plots of the round key sequence show the randomness in each case of plain text and password pair.

By FMEA a pretty good one-to-many mapping between plain text and cipher text characters can be obtained. One-to-many mapping is needed to reduce the possibility of cracking any cipher by observing its frequency description. The best known polyalphabetic cipher 'Vigenere cipher' [1, 8] is good to provide on this need but it still has some limitations. For example, if any analyst can somehow determine the length of the keyword that is used to encrypt any plain text which has enough length and good number of repetitions of characters, then Vigenere cipher may also have repetitions in its cipher.

Example 2: In this example, a plain text is encrypted using Vigenere Cipher algorithm (for keyword DECEPTIVE) and repetitions of characters are shown using shaded cells in Table 3. In this case, VTW is repeated twice in the cipher text for the same plaintext repetition of RED and for this the portion of the key is EPT which is also repeated. This repetition can be avoided if the keyword's length is large enough. This repetition in the Vigenere cipher aids an analyst in predicting the length of the keyword. When the length is known, the keyword can be found by observing the frequency description of the cipher. These limitations in Vigenere cipher can be overcome by using FMEA. The reason is that, the polyalphabetic round key generated by

FMEA is highly random in nature of sequence. FMEA is made as it can generate extremely non-periodic and unpredictable round key sequence. Moreover this sequence is dependent not only on the password itself but also on the plain text. So it can provide a highly secure key sequence.

Table 2. Randomness of the proposed technique

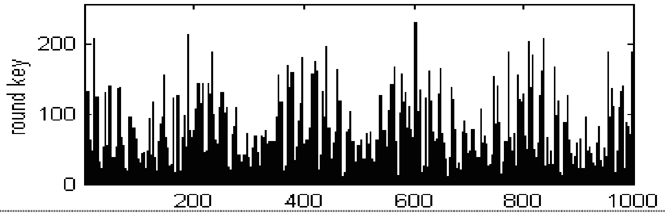
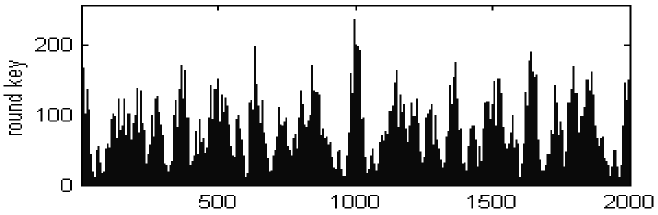
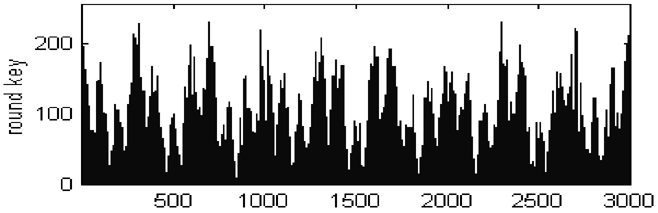
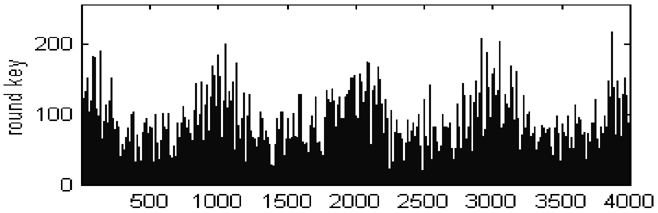
***L* (Length of the Plaintext, Generated Randomly)**	**Password**	**Round Key Sequence**
1000	*12abc782*	
2000	*Faithful*	
3000	*Gotohell*	
4000	*abccbaxyz*	

Table 3. Ciphertext for a given plaintext in Vigenere Cipher (for keyword DECEPTIVE)

Key	D	E	C	E	P	T	I	V	E	D	E	C	E	P	T	I	V	E	D	E	C	E	P	T	I	V	E
Ptext	W	E	A	R	E	D	I	S	C	O	V	E	R	E	D	S	A	V	E	Y	O	U	R	S	E	L	F
Ctext	Z	I	C	V	T	W	Q	N	G	R	Z	G	V	T	W	A	V	Z	H	C	Q	Y	G	L	M	G	J

Table 4. Ciphertext for a given plaintext in FMEA (for keyword DECEPTIVE)

Ptext	W	E	A	R	E	D	I	S	C	O	V	E	R	E	D	S	A	V	E	Y	O	U	R	S	E	L	F
Ctext	j	p	Q	X	a	c	k	@	E	S	a	□	d	t	R	Q	Q	E	]	S	L	G	M	p	d	Y	I

Example 3: The plain text in the Example 2 is encrypted with password DECEPTIVE using FMEA and presented in Table 4. It is seen that there is no repetition in the FMEA cipher text for the repetition of RED in the plain text. FMEA key sequence is a pretty good random sequence that can also be used in other cryptographic applications. For example this can be used to produce pseudorandom numbers which is a very important tool in cryptographic application like authentication. In the classical pseudorandom number generators, modular arithmetic with large prime numbers are used. As FMEA doesn't use modular arithmetic but still well in generating random sequence, it can easily be used in such applications where computational complexity must be reduced. FMEA can be used effectively for the authentication technique proposed in [9].

6. Conclusion

We propose an algorithm Fourier Masking Encryption Algorithm (FMEA) which is a polyalphabetic symmetric key encryption. The algorithm not only possesses one-to-many mapping between plaintext and cipher text, but also establishes diffusion, confusion and avalanche among plaintext, password and cipher text. The proposed algorithm is also easy to implement in hardware, mobile phones, PDA as it doesn't require any extra storage capacity. The efficiency of the proposed algorithm over the existing algorithms is shown in experimental results which show the superiority of the proposed algorithm.

References

[1] William Stallings, *Cryptography & Network Security (Principles & Applications)*, 3rd Edition, 2003.

[2] S. G. Telang, *Number Theory*, First Edition, 1996.

[3] Web-link: http://mathworld.wolfram.com/FourierSeries.html.

[4] Web-link: http://en.wikipedia.org/wiki/Fourier_series.

[5] John G. Proakis, Dimitris G. Manolakis, *Digital Signal Processing (Principles, Algorithms & Applications)*, 3rd Edition, Prentice Hall India, 2003.

[6] Simon Haykin, Barry Van Veen, *Signals and Systems*, John Wiley & Sons, Inc. 1999.

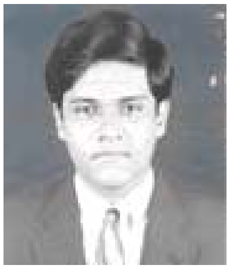
[7] Winograd, S., *On Computing the Discrete Fourier Transform*, Math. Comp., Vol. 32, pp. 177-199.

[8] Web-link: http://www.trincoll.edu/depts/cpsc /cryptography/vigenere.html.

[9] M. L. Rahman, S. Rafique, M. I. Jabiullah, S. M. M. Rahman, "Strong Authentication Using Pseudo-Random Bit Stream", *Dhaka University Journal of Science,* 54 (1) 59-61 2006 (January).

Nadim Jahangir is a first year undergraduate student of Computer Science and Engineering, Northern University, Bangladesh.

His research interests include Cryptography and Network Security, Robot Vision, Fuzzy Control Systems, Data Communication and Computer Networks, Mobile Computing. He is working on a project for introducing the capability of controlling remote PC by mobile phones in world-wide manner.



Ahsan Raja Chowdhury received his B.Sc. (Hons) and MS degrees in Computer Science and Engineering from University of Dhaka, Bangladesh.

He is now in the Department of Computer Science and Engineering, Northern University, Bangladesh as a Senior Lecturer and recently joined in the Department of Computer Science and Engineering, University of Dhaka as a part time Lecturer. He has participated in several National and International conferences for presenting his research papers. His research interests include Image Processing, Cryptography and Network Security, Logic Synthesis and Design, Reversible Logic, Fuzzy Logic.