A Novel Text Encryption Algorithm based on Core Adaptive Fourier Decomposition

Lei Dai
Faculty of Science and Technology
University of Macau
Macao China
yb87427@um.edu.mo

Zhijing Ye
School of Science
Wuhan University of Technology
Wuhan China
xkinghust@163.com

Liming Zhang
Faculty of Science and Technology
University of Macau
Macao China
lmzhang@um.edu.mo

Tao Qian
Macau University of Science and Technology
Macao China
tqian@must.edu.mo

ABSTRACT

The progress of modern science and technology has accelerated the development of communication, providing more focus on the security of data transmission. Although there are many recognized encryption techniques, more high-performance algorithms are required to withstand the sophisticated analysis and cracking techniques. In this paper, we propose an innovative algorithm that combines with Core Adaptive Fourier Decomposition (Core-AFD) for text encryption. In addition, two versions of algorithm, initial version and enhanced version, are discussed for different forms and strength of the secret key and ciphertext. Due to the one-to-one correspondence between the decomposition and the error terms, the encryption algorithm is theoretically feasible and safe. The algorithm is further demonstrated by conducting experiments on text encryption with different lengths. The results are promising. Finally, the time complexity of cracking our algorithm in different cases is given.

CCS CONCEPTS

• Security and privacy → Software and application security → Domain-specific security and privacy architectures • Security and privacy→ Cryptography → Information-theoretic techniques

KEYWORDS

Cryptography, Text Encryption, Adaptive Fourier Decomposition, Signal Processing

ACAI '19, December 20–22, 2019, Sanya, China

ACM ISBN 978-1-4503-7261-9/19/12…$15.00
https://doi.org/10.1145/3377713.3377798

ACM Reference format:

Lei Dai, Zhijing Ye, Liming Zhang and Tao Qian. 2019. A Novel Text Encryption Algorithm based on Core Adaptive Fourier Decomposition. In *Proceedings of 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI'19). Sanya, China, 7 pages.* https://doi.org/10.1145/3377713.3377798

1 Introduction

Cryptography converts important plain text into unreadable ciphertext to prevent intruders from intercepting them. It used to involve the diplomatic and military fields, but due to the rapid development of network technology, it has recently been applied to many areas of real life. Various encryption approaches using different techniques have been proposed over the years. Some classic and effective encryption algorithms (Data Encryption Standard, DES; Advanced Encryption Standard, AES and Rivest-Shamir-Adleman, RSA [1]-[3]) are widely used in the cryptography area. Since 1990s, some researchers have found that maybe one can establish relationships between cryptography and chaos because chaos have the characteristics of non-periodicity, randomness and sensitivity to initial conditions [4]. A plenty of encryption methods using chaos have been developed.

In the field of signal analysis, the positive frequency representation of signal is always a hot topic. Based on the theory of mono-component function, a series of Adaptive Fourier Decomposition (AFD) methods have been invented, including Core-AFD, Unwinding AFD and Cyclic AFD [5]-[7]. They have been applied in many aspects like ECG signal classification, image reconstruction and system identification because of its fast convergence which has been proved.

In this paper, we discuss how to combine encryption with signal decomposition method and propose a novel text encryption algorithm based on Core-AFD. Relations of ciphertexts to both the plaintext and secret key should be weakened in an excellent encryption method. As the feature of one-to-one correspondence between decomposition and error term, it would be feasible and

effective to set them as the cyphertext and secret key respectively. Taking encryption strength into consideration, two versions of our encryption algorithms are displayed. The difference is that there is a variable parameter in the enhanced version, making it more secure and harder to be attacked.

The rest of this paper is organized as follows. Section 2 gives a brief introduction of Core-AFD. Our proposed encryption algorithm in two versions and the decryption process are presented in Section 3. Section 4 displays the experimental results of our algorithm. In Section 5, the security of our method is analyzed. Section 6 shows conclusions.

2 Preliminaries

Motivated by research interest on positive instantaneous frequency of signals, a series of AFD algorithm has been developed. AFD obtains the positive frequency expansion of analytic rational function in the case of simple and complex variables functions. Additionally, its fast convergence is also proved.

2.1 Takenaka-Malmquist system [8]

The Takenaka-Malmquist system is also called the rational orthogonal system, which is an important component of AFD. Such rational fractions are fundamental modules to rational functions in Hardy spaces H^2. In complex plane $\mathbb{C}$, $\{B_k\}_{k=1}^{\infty}$ the Takenaka-Malmquist system generated by $a_k \in \mathbb{D}$, where $\mathbb{D}$ denotes the unit disc, is defined by

$$B_k(z) = B_{\{a_1,\cdots,a_k\}}(z) = \frac{\sqrt{1-|a_k|^2}}{1-\bar{a}_k z}\prod_{l=1}^{k-1}\frac{z-a_l}{1-\bar{a}_l z}. \quad (1)$$

The Takenaka-Malmquist system consists of functions with positive frequencies which can be seen from (1).

It has been demonstrated with long-term studies that the system is complete in disc algebra $A(\mathbb{D})$ and Hardy p spaces $H^p(\mathbb{D})$, $1 \le p \le \infty$, if and only if the below Szaśz condition satisfies (see [9], [10])

$$\sum_{k=1}^{\infty}(1-|a_k|) = \infty. \quad (2)$$

2.2 Core-AFD [5]

As opposed to traditional Takenaka-Malmquist system, Core-AFD holds excellent adaptability, providing an approach to decompose complex-value analytic functions in complex H^2 and general real-valued functions in Lebesgue L^2 spaces. This paper concentrates on real-valued functions.

Denote by F the real-valued function with finite energy on the unit circle ∂D. Since it can be divided into F^+ and F^- in the form of direct sum decomposition

$$\begin{aligned} F(e^{it}) &= \sum_{k=-\infty}^{\infty}\rho_k e^{ikt} \\ &= \sum_{k=0}^{\infty}\rho_k e^{ikt} + \sum_{k=-\infty}^{-1}\rho_k e^{ikt} \\ &= F^+(e^{it}) + F^-(e^{it}), \end{aligned} \quad (3)$$

where $\rho_k = \frac{1}{2\pi}\int_0^{2\pi} F(e^{it})e^{-ikt}d\,t$ holds the equation that $\rho_{-k} = \bar{\rho}_k$, then there is

$$F(e^{it}) = -\rho_0 + 2\mathrm{Re}F^+(e^{it}). \quad (4)$$

Here F^+ is the projection of F onto H^2. Therefore, the problem can be reduced to the decomposition of F^+.

2.2.1 *Szegö kernel [11]*

Core-AFD introduces unit module of the unit disc, Szegö kernel e_a, whose linear combination is contained in H^2.

$$e_a(z) = \frac{\sqrt{1-|a|^2}}{1-\bar{a}z}, a \in \mathbb{D}. \quad (5)$$

Drawing lessons from back shift operator, we rewrite F^+ as follows

$$\begin{aligned} F^+(z) = F_1^+(z) &= \langle F_1^+, e_{a_1}\rangle e_{a_1}(z) + \frac{F_1^+(z) - \langle F_1^+, e_{a_1}\rangle e_{a_1}(z)}{\frac{z-a_1}{1-\bar{a}_1 z}}\frac{z-a_1}{1-\bar{a}_1 z} \\ &= \langle F_1^+, e_{a_1}\rangle e_{a_1}(z) + R_1(z), \end{aligned} \quad (6)$$

where a_1 can be any complex number in unit disc. Denote by $F_2^+(z) = \frac{F_1^+(z) - \langle F_1^+, e_{a_1}\rangle e_{a_1}(z)}{\frac{z-a_1}{1-\bar{a}_1 z}}$, then there holds

$$R_1(z) = F_2^+(z)\frac{z-a_1}{1-\bar{a}_1 z}. \quad (7)$$

So (6) is identical with the below equation (8)

$$F^+(z) = \langle F_1^+, e_{a_1}\rangle e_{a_1}(z) + F_2^+(z)\frac{z-a_1}{1-\bar{a}_1 z}. \quad (8)$$

Repeating the process until $F_n^+(z)$ by recursive formula

$$F_{k+1}^+(z) = \left(F_k^+(z) - \langle F_k^+, e_{a_k}\rangle e_{a_k}\right)\frac{1-\bar{a}_k z}{z-a_k}, \quad (9)$$

we will obtain the decomposition formula

$$F^+(z) = \sum_{k=1}^{n}\langle F_k^+, e_{a_k}\rangle B_k(z) + F_{k+1}^+(z)\prod_{k=1}^{n}\frac{z-a_k}{1-\bar{a}_k z}, \quad (10)$$

where the remainder term after *k*-th decomposition is expressed as

$$R_k(z) = F_{k+1}^+(z)\prod_{k=1}^{n}\frac{z-a_k}{1-\bar{a}_k z}. \quad (11)$$

2.2.2 *Maximal Selection Principle (MSP) [12]*

In addition to the desire to construct a positive instantaneous frequency decomposition system with an explicit expression like (10), it is also eager for Core-AFD that the instantaneous frequency be increased in accordance with the decomposition hierarchy. That is the reason it applies MSP to select applicable $a_k, k = 1,\cdots,n$.

Szegö kernel is the Cauchy kernel in arc length measure so that it has the nature of reproducing kernel [13]. It is due to the orthogonality of Hilbert space and unit modular property of complex number for Möbius transformation [14], we can derive the below energy relation from (8)

$$|F^+|^2 = |\langle F_1^+, e_{a_1}\rangle e_{a_1}|^2 + |F_2^+|^2 = (1-|a_1|^2)\left|F_1^+(a_1)\right|^2 + |F_2^+|^2. \quad (12)$$

To extract the maximum energy from the first decomposition $\langle F_1^+, e_{a_1}\rangle e_{a_1}(z)$, our purpose can be converted to maximize $(1-|a_1|^2)\left|F_1^+(a_1)\right|^2$ in the range of unit disc $\mathbb{D}$, we have

$$a_1 = \arg\max\left\{(1-|a|^2)\left|F_1^+(a)\right|^2 : a \in \mathbb{D}\right\}. \quad (13)$$

We perform this maximum selection in a similar way to get all the befitting a_k for (10)

$$a_k = \arg\max\left\{(1-|a|^2)\left|F_k^+(a)\right|^2 : a \in \mathbb{D}\right\}. \quad (14)$$

The decomposition process is a consecutive energy approximation which means in the L^2-norm sense,

$$F^+(z) = \sum_{k=1}^{\infty} \langle F_k^+, e_{a_k}\rangle B_k(z). \quad (15)$$

For a proof, it can refer to [15] and [16].

3 Proposed Encryption Algorithm

Our proposed text encryption algorithm adopts a novel thought of signal decomposition, portending it quite different from existing encryption methods, nonetheless, its encryption performance not compromised.

The two most crucial parts of cryptography are the secret key and ciphertext. It is critical to choose the secret key and ciphertext appropriate for Core-AFD so that it could not only meet the fundamental characteristics, but also can guarantee a splendid encryption and decryption property. Inspired by the relationship, one-to-one correspondence, between error and decomposition term, we found the feasibility to employ them as the secret key and ciphertext respectively. The flowchart of two versions encryption process is shown in Fig.1.

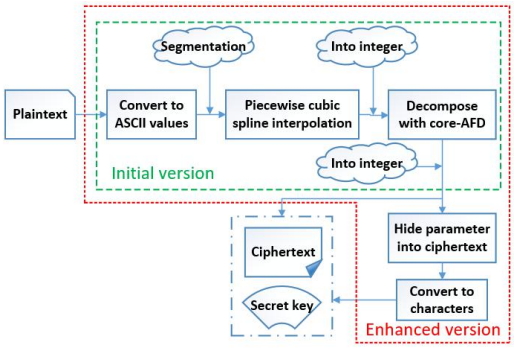

Figure 1: The flowchart of two versions encryption process: 1) initial version in green box; 2) enhanced version in red box.

3.1 Re-processing

The common operation to process characters in text files is to convert them into ASCII values, which implies the targets we will face become a series of integers with distinct numerical difference. There are risks and costs if integers of this type are processed directly with Core-AFD. Since almost every integer forms a peak or valley in the curve, it will need to increase the number of the decomposition for anticipated outcomes. Otherwise, the error will be large enough to affect encryption and decryption. Thus, smoothing is required before applying Core-AFD to converted integers.

There are plenty of smoothing methods available and may have a positive impact on our encryption algorithm. In this paper, we select piecewise cubic spline interpolation as the smoothing method for the listed five reasons:

(1) Calculation is simple and easy to implement;

(2) Function is continuous;

(3) Generate smooth curve;

(4) Curvature changes continuously;

(5) Degree of polynomial is low.

3.2 Encryption Algorithm

Denote by T the converted ASCII values of the plaintext, N the length of the whole plaintext. Considering the computation of long text processing, segmentation for the plaintext is introduced. Set η as an appropriate threshold for the length of text. Then we will obtain the subsection texts for encryption, expressing as

$$\widetilde{T} = \begin{cases} T_i, N > \eta \\ T, N \le \eta \end{cases}, \quad (16)$$

where $i = 1,\cdots,m$, $m = \lceil N/\eta \rceil$. The length of $T_k, k = 1,\cdots,m-1$ is $\lfloor N/m \rfloor$. Here, symbol '$\lceil\ \rceil$' means rounding up to an integer and symbol '$\lfloor\ \rfloor$' expresses to round to the nearest integers towards minus infinity.

As mentioned in Section 3.1, the piecewise cubic spline interpolation is executed for the $\widetilde{T}$ to acquire smooth and continuous function, denoting as f after rounding each value to the nearest integer. After that, decompose real integer function f in the form of Core-AFD with proper number of decomposition M. The partial sum based on projection $f^+ \in H^2$ of f can be decomposed as

$$\tilde{f}^+(z) = \sum_{k=1}^{M} \langle f_k^+, e_{a_k}\rangle B_k(z), \quad (17)$$

then we will obtain

$$\tilde{f} = -\rho_0 + 2\mathrm{Re}\tilde{f}^+, \quad (18)$$

where $\rho_0 = \frac{1}{2\pi}\int_0^{2\pi} f(e^{it})dt$, and a_k is ensured by MSP.

3.2.1 Initial Version

In order to correspond to ASCII values, ranging from 0 to 126 in decimal numeral system, it is required to round up every value of $\tilde{f}$ to an integer. The rest points of the smoothed function are recorded as the ciphertext $C^{(1)}$ when removing all the interpolation ones. The initial secret key $K^{(1)}$ at this encryption consists of all non-interpolation points of $f - \lceil \tilde{f} \rceil$. In this initial encryption version, both the secret key and ciphertext are presented as integers. They are not converted into characters by ASCII because there are probably some values not between 0 and 126. Our proposed text encryption algorithms of initial version based on Core-AFD is as Algorithm 1.

Algorithm 1: Initial version

Input: plaintext, length of the plaintext N, segmentation threshold η and number of the decomposition M.

Output: secret key $K^{(1)}$ and ciphertext $C^{(1)}$.

1: Convert the plaintext into corresponding ASCII values T.

2: **if** $N > \eta$ **do**

3: Segment the plaintext, $\widetilde{T} = T_i, i = 1,\cdots,\lceil N/\eta \rceil$;

4: **else** No segmentation, $\widetilde{T} = T$;

5: end if

6: Execute piecewise cubic spline interpolation for $\widetilde{T}$;

7: Round values of smoothed function to nearest integer, denoted by f;

8: Decompose f with Core-AFD:

Get $\tilde{f}^{+}(z) = \sum_{k=1}^{M} \langle f_k^{+}, e_{a_k} \rangle B_k(z)$;

Get $\tilde{f} = -\rho_0 + 2\mathrm{Re}\tilde{f}^{+}$;

9: **return** $C^{(1)}$ from $\lceil \tilde{f} \rceil$ and $K^{(1)}$ from $f - \lceil \tilde{f} \rceil$.

3.2.2 Enhanced Version

However, further process to initial secret key is considered for visualization, at the same time, improving encryption performance and difficulty of decryption. We introduce β which ensures all values of $K^{(2)}$ are ASCII values within 0 and 126, to obtain every $K_i^{(2)}$ of the new secret key $K^{(2)}$.

$$K_i^{(2)} = K_i^{(1)} + \beta - \min_{1 \le k \le N} K_k^{(1)}. \tag{19}$$

As $\beta - \min_{1 \le k \le N} K_k^{(1)}$ is necessary in decryption process, it is hidden in the ciphertext, forming the new ciphertext $C^{(2)}$.

$$C^{(2)} = \{\beta - \min_{1 \le k \le N} K_k^{(1)}, C^{(1)}\}. \tag{20}$$

Then we can make the secret key and ciphertext appear as characters as possible by transforming them with ASCII. To simplify, still denote them by $K^{(2)}$ and $C^{(2)}$. The enhanced version of our algorithm is as Algorithm 2.

Algorithm 2: Enhanced version

Input: plaintext, β, length of the plaintext N, segmentation threshold η and number of the decomposition M.

Output: secret key $K^{(2)}$ and ciphertext $C^{(2)}$.

1: Convert the plaintext into corresponding ASCII values T.

2: **if** $N > \eta$ **do**

3: Segment the plaintext, $\widetilde{T} = T_i, i = 1,\cdots,\lceil N/\eta \rceil$;

4: **else** No segmentation, $\widetilde{T} = T$;

5: end if

6: Execute piecewise cubic spline interpolation for $\widetilde{T}$;

7: Round values of smoothed function to nearest integer, denoted by f;

8: Decompose f with Core-AFD:

Get $\tilde{f}^{+}(z) = \sum_{k=1}^{M} \langle f_k^{+}, e_{a_k} \rangle B_k(z)$;

Get $\tilde{f} = -\rho_0 + 2Re\tilde{f}^{+}$;

9: Get $C^{(1)}$ from $\lceil \tilde{f} \rceil$ and $K^{(1)}$ from $f - \lceil \tilde{f} \rceil$;

10: Get $K_i^{(2)} = K_i^{(1)} + \beta - \min_{1 \le k \le N} K_k^{(1)}$;

Get $C^{(2)} = \{\beta - \min_{1 \le k \le N} K_k^{(1)}, C^{(1)}\}$;

11: **return** $C^{(2)}$ and $K^{(2)} = \{K_i^{(2)}\}, i = 1,\cdots,N$.

3.3 Decryption Process

The principle of decryption is simple when grasping the encryption process well. We will display the decryption methods of two encryption versions respectively.

- Decryption of initial version

In the initial encryption algorithm, the secret key and ciphertext are presented as integers rather than characters. They correspond to the decomposition term and the error term, whose lengths are the same as that of the plaintext. It is just a simple sum that will restore the original signal. Then the decrypted plaintext $\dot{T}$ is obtained as

$$\dot{T} = C^{(1)} + K^{(1)}. \tag{21}$$

- Decryption of enhanced version

Although it will be slightly complex when proceeding the decryption for enhanced encryption algorithm, the security of enhanced version is higher. By default, both sides of communication know where $\beta - \min_{1 \le k \le N} K_k^{(1)}$ is placed in the ciphertext so that it can be extracted to make sure the length of the ciphertext consistent with the plaintext and secret key. It is noticed that $C^{(1)}$ is the value after removing $\beta - \min_{1 \le k \le N} K_k^{(1)}$ from

the ciphertext. We will then obtain decrypted plaintext $\dot{T}$ which should be

$$\dot{T} = C^{(1)} + K^{(2)} - (\beta - \min_{1\le k\le N} K_k^{(1)}). \quad (22)$$

4 Experimental Results

In this section, we will respectively conduct experiments based on our encryption algorithm in two versions and compare the results with some classical algorithms, like Rivest-Shamir-Adleman (RSA), Advanced Encryption Standard (AES) and hybrid AES and Data Encryption Standard (hybrid AES-DES). The performance of all the algorithms performed in texts with different lengths are compared.

For initial version, we set the segmentation threshold as 100 and the number of the decomposition as 50. And β in enhanced version is set as 33 because characters from 32 to 126 are visible symbols in ASCII codes and 32 is space. The comparison results are shown in Table 1. All the experiments are carried out using Matlab R2016a version in a desktop whose CPU is Intel Core i7-6700, clock frequency is 3.4GHz and static power is around 109W.

Table 1. Comparison results of different algorithms. The bold time respectively represents the shortest encryption time and decryption time within five algorithms when encrypting different character counts.

Method	Character count	Encryption time(s)	Decryption time(s)
RSA	100	**0.011**	0.0078
	634	**0.030**	0.0122
	2004	**0.045**	0.0390
	4133	**0.047**	0.0391
	6287	**0.158**	0.0560
	8270	**0.191**	0.0549
	10088	**0.237**	0.0623
	20801	**0.401**	0.2325
AES	100	0.805	**0.0004**
	634	1.298	0.0025
	2004	2.976	0.0092
	4133	6.185	0.0132
	6287	5.936	0.0178
	8270	6.686	0.0183
	10088	8.128	0.0264
	20801	16.07	0.0471
Hybrid AES-DES	100	0.755	0.0543
	634	1.298	0.0026
	2004	2.898	0.0079
	4133	16.18	0.0092
	6287	20.67	0.0133
	8270	26.86	0.0535
	10088	32.72	0.0714
	20801	67.10	0.1016
Our initial version	100	0.322	0.0020
	634	1.390	**0.0024**
	2004	3.833	**0.0049**
	4133	7.131	0.0087
	6287	11.65	0.0117
	8270	15.34	0.0164
	10088	17.57	0.0235
	20801	35.57	0.0385
Our enhanced version	100	0.252	0.0020
	634	1.181	0.0028
	2004	2.979	0.0051
	4133	5.925	**0.0059**
	6287	8.294	**0.0061**
	8270	11.01	**0.0058**
	10088	13.26	**0.0060**
	20801	25.45	**0.0065**

Table 2. One example for the encryption and decryption results of two versions. The rectangles in the ciphertext of enhanced version are some characters whose values are within 0-31, resulting in no corresponding visible symbols.

Plaintext	Notice [1]:I need you to help me encrypt this text and hide the time, place, characters and things in it.Then you need to send the encrypted information to Mr. Zhang. Company MathWorks most complete software for computational computer produces; the main program the company that actually Passport is software MATLAB (' short for Mat rix Lab Oratory and means lab Matrix') is one of the most advanced software, algorithms and math and a programming language developed generation fourth is possible. Acknowledgements are usually put in the end of the paper, between the reference and the content. There should be no number in this part.	
Initial Version (Ciphertext, secret key)	82, 107, 103, 104, 95, 108, 52, 31, 43, 71, 65, 40, 22, 112, 106, 115, 110, 5, 121, 122, 113, 15, 121, 97, 5, 123, 101, 124, 114, 6, 124, 106, 26, 104, 118, 115, 114, 112, 110, 118, 57, 119, 112, 103, 112, 28, 109, 118, 120, 110, 17, 97, 110, 112, 29, 108, 100, 107, 105, 27, 118, 110, 101, 15, 116, 107, 100, 59, 18, 40, 112, 122, 106, 103, 99, 32, 19, 101, 106, 110, 110, 109, 103, 107, 105, 117, 106, 29, 90, 118, 57, 38, 104, 97, 109, 113, 113, 102, 22, 105, 114, 17, 110, 101, 99, 78, 106, 109, 95, 7, 118, 111, 122, 37, 109, 95, 103, 93, 18, 116, 120, 57, 107, 102, 101, 92, 15, 110, 102, 51, 21, 124, 124, 55, 123, 124, 122, 120, 117, 105, 19, 22, 114, 117, 117, 118, 118, 115, 119, 119, 120, 116, 25, 116, 111, 1, 50, 126, 16, 47, 96, 124, 115, 95, 113, 52, 19, 25, 74, 102, 111, 114, 114, 112, 125, 25, 62, 122, 116, 114, 52, 121, 114, 111, 124, 106, 119, 122, 120, 26, 104, 115, 117, 116, 114, 110, 116, 95, 42, 122, 126, 92, 116, 124, 94, 118, 100, 24, 92, 122, 106, 23, 95, 113, 117, 119, 120, 59, 103, 114, 118, 114, 111, 110, 40, 96, 114, 115, 113, 109, 104, 117, 20, 117, 122, 116, 91, 114, 102, 100, 117, 62, 22, 96, 103, 96, 3, 111, 106, 105, 99, 35, 118, 105, 109, 111, 110, 100, 102, 21, 112, 120, 109, 42, 52, 113, 112, 122, 96, 110, 119, 24, 119, 98, 105, 120, 95, 3, 111, 116, 122, 96, 107, 114, 116, 22, 79, 101, 103, 113, 116, 119, 121, 114, 21, 119, 124, 20, 106, 122, 109, 113, 120, 105, 108, 104, 32, 89, 72, 75, 73, 65, 88, 42, 39, 25, 122, 104, 119, 118, 114, 30, 106, 116, 17, 16, 97, 105, 123, 32, 100, 117, 114, 37, 71, 109, 103, 46, 92, 104, 109, 110, 102, 121, 121, 26, 95, 112, 107, 32, 100, 107, 96, 107, 114, 91, 95, 93, 97, 27, 73, 105, 113, 118, 122, 119, 26, 44, 35, 105, 115, 111, 22, 109, 96, 21, 113, 109, 43, 121, 121, 103, 33, 104, 120, 125, 125, 32, 107, 103, 106, 108, 103, 105, 95, 93, 25, 116, 116, 112, 122, 121, 112, 119, 112, 24, 28, 90, 97, 102, 105, 106, 107, 107, 108, 110, 113, 28, 90, 52, 87, 41, 105, 104, 121, 32, 28, 110, 71, 94, 28, 100, 32, 124, 114, 114, 113, 114, 115, 116, 116, 117, 122, 92, 19, 108, 115, 104, 99, 102, 100, 103, 96, 34, 94, 105, 108, 109, 109, 109, 103, 104, 93, 26, 113, 123, 107, 103, 109, 110, 110, 110, 111, 110, 28, 105, 115, 120, 121, 123, 116, 33, 109, 112, 27, 116, 115, 114, 114, 113, 87, 116, 110, 58, 3, 1, 96, 106, 112, 117, 121, 111, 102, 105, 107, 108, 109, 111, 112, 113, 116, 41, 104, 125, 100, 20, 124, 121, 120, 104, 111, 114, 113, 96, 102, 111, 121, 21, 106, 104, 37, 114, 100, 101, 32, 106, 116, 99, 22, 107, 101, 38, 117, 111, 97, 115, 118, 105, 105, 104, 116, 28, 38, 96, 108, 112, 19, 101, 103, 115, 35, 121, 111, 93, 30, 109, 101, 104, 105, 109, 105, 107, 103, 97, 32, 115, 115, 111, 43, 113, 110, 94, 26, 94, 111, 115, 119, 9, 117, 106, 49, 35, 59, 107, 105, 120, 100, 22, 113, 107, 113, 116, 116, 96, 41, 97, 99, 45, 107, 116, 23, 113, 123, 113, 107, 104, 116, 20, 111, 115, 29, 121, 106, 113, 115, 25, 119, 101, 113, 115, 45.	-4, 4, 8, 1, 1, -7, -20, 10, 6, 22, -7, 32, 0, -2, -5, -14, -10, 24, 0, 11, 4, 14, -5, 14, 27, -19, 0, -16, -2, 26, -15, -5, 6, -3, -5, -16, 9, 9, 2, -2, -25, -2, -9, 2, 3, 4, 7, -17, 0, 6, 15, 0, 0, -12, -7, -4, 5, -7, -4, 5, -2, -6, 0, 14, 0, -2, 9, 13, 26, -5, 0, -15, -9, -4, 2, 12, 13, -2, -2, -13, 4, -12, -9, 9, -7, -7, 9, -7, 7, -8, 13, -6, 12, 7, -4, -2, -10, 12, 10, 0, -4, 15, -5, 15, -13, 5, -2, -8, 15, -8, 3, 0, -5, -5, 1, 6, -7, 12, 14, 0, -9, -25, 8, -2, 9, 8, 14, 6, 2, 20, 1, -23, -14, 11, -11, -2, -10, -4, -16, -5, 13, 3, -4, -15, -6, -4, -9, -21, -2, -14, -9, -6, -3, 0, 0, 31, 27, -12, 30, -15, -6, -20, -15, 15, -10, -6, -9, -15, -7, 9, -2, -2, -17, -2, -4, 15, -25, 0, -10, -10, 0, -4, -9, 16, 3, -8, -7, -4, 6, -5, -4, -8, -4, -6, -9, 0, 6, -11, -7, -15, 9, 0, -5, 3, -4, 1, 8, 10, -11, 8, 9, 1, -2, -8, -7, -1, -2, -6, 2, -10, -2, -2, -14, -2, -11, 2, -2, -7, -3, 4, 7, -2, -2, 2, -8, -5, 9, 3, -2, 1, -2, -3, 0, 20, 1, 5, -11, -2, -9, -3, 1, -3, -6, 2, -8, 4, -2, 7, 11, 4, -16, -7, -10, 16, -2, -2, -10, 11, 0, 2, 8, -11, -4, -4, 8, -12, 0, -5, 1, 1, -6, 5, 0, 1, -4, 7, 2, -4, -5, -7, 2, 1, -14, -9, 12, 9, -11, -7, 2, -1, -3, 5, -2, 0, -12, -7, 9, 3, 3, -10, 1, 14, -7, 0, -8, -4, 2, 2, -4, -5, -2, 16, -10, -3, -7, 0, 14, -12, 6, -5, 5, -12, -5, -14, -2, 10, -12, 6, 9, -7, 0, 6, 2, -2, -7, -1, 9, -6, 1, 3, 1, 1, 12, 4, 1, 3, 4, -3, 3, -4, -17, 1, 13, -2, -3, 0, 2, 1, -11, 1, 5, 11, -7, -7, -11, -5, -17, -2, -6, 5, -9, -10, -9, -10, -2, -12, -11, 2, -9, -4, 7, 7, -10, -5, -10, -6, 2, -16, -5, -11, 20, 4, 7, 11, 1, 6, 9, -2, 9, -4, -1, 2, 4, 7, -13, 13, -9, 4, -7, -5, 22, -4, -13, -7, 6, 4, -3, 0, -12, 0, -2, -10, 0, -18, -7, -7, -12, 12, 11, 13, 0, -15, 6, 4, 15, -3, 0, 5, -2, 6, -4, 10, -8, -1, 4, -2, 7, 6, -10, -2, 3, -7, 5, -13, 6, -5, 0, 0, -6, -6, -7, -3, -7, -7, -12, 1, -4, 2, 5, -4, -4, 1, 1, -5, 11, -5, -9, -12, 7, -6, 3, 1, -2, -6, -2, -2, -5, -4, -7, 0, -10, -2, -2, -1, -9, -7, -11, 1, 2, 7, -6, -3, -7, -2, -6, 8, -4, 10, 6, -5, 11, -1, 6, -5, 2, 4, 0, 0, -5, -6, 2, 0, 4, 1, -6, -1, -7, 4, 17, -6, -9, 7, -3, -2, 6, -3, 2, -7, 4, 0, 0, -2, -5, -2, -5, -7, 3, 2, 5, 0, -2, -7, 5, -7, 3, -4, 4, 0, -19, -3, -11, -11, 2, -6, 7, -4, 5, 0, -5, -2, 2, -7, 10, -3, -3, -5, -2, -4, -6, 1, 10, 2, 0, -2, 1, -3, -6, -9, 1, 2, -13, 3, -5, 9, -3, -6, -4, -9, -3, -2, 12, -6, -5, 3, -8, -2, -8, 0, 7, -7, -4, 1, -2, 1.	

Enhanced Version (Ciphertext, secret key)	<Rlmibm2R)HC* qhvs psq□ya□{e}t□| f□nqpponx7vtqh□rv|1□bml %gg ROpoi□t1fY□&p}qfU□:Soqponmmui'YwY (|kprqh□kp□ot3Jdjf ry {□fooV□wv08jh_□pkY:y {syzwvsz□fqtuuvvvwvz□to 1}□-_h| uhj□[clprpd4BxtpSzyet□guyw□grttrnt" tZ`.t|_vd□]y {□`qvxyyesutso|.'qssqniuuzuYuie`5 ]JP'spar□mvwvq`g□rto2UjkpYny:hbmv"Xmzmenrs Qelpruvpss□sxkjkmpi"ALNLG<*&□yhxvriut□Xhz|brv)Ylf- Pgnqssn□dpm□af_kj5d2c□Khpuzx□-"ib*}ja□|f stb"ftxzSWfjlmlj]□luwwxxwk□! D|eilmopsq#fvY (ir1U□luY3]zrrqstuvvsi□scidhef` $_kmmnmlhS□pgkmnnmnno&lvwxst! pa□usrqpXsm9□G`iotypghiklmpwr'e\|d wxpzuuy$uks*1f'msd□nu_□ei$uqbcmnld) `lopppnc }la nfh111je$□nvs,wma2`lopprt.&Wkjxd□luvvsh `ac.lr□twrmgu□ps{pws□weou4	3?C<-4)EDQ3[(99+-ZFS@L7JU)< +SX-;F39/>F)8%:04GU>+SDK;=47=>Q0.Q5SN<9CHW6< +,9LP'L5,>.1C49FSD3G4S9S82GG:.N6<7F@?D<C26WD22JK:5,\7BAjQ9H"(/%5=5:.9G7 9.692':.4SN<(X1W/7<!5;0.)6H=<+:Q\(G k<4@15B:HB669C995963<C16-)<7>S=DE23C?:34ST8=077.,0A:69@B39? 746G<6<0B<S:Q56-+SM?84- +=BI>02*jA>(D<("HBO::E26DS:6B<;SC?:6 3@?2<F<37FHO>S:H1B<6E2>H6<3S>?)6:K156;L1>3/1$/;G/?S;GT9:3DH;)? $'DC;C>5@9+<J3:<MP.@@K/<S6O?:C776$F: fi=37CCDE+9:%76M;YX>EB8A46>974G4@ +DOU15G)@>2<92.)43/7:L5.A? I5=ABAEB4:>@9MB2:? 4A/B8=:665:660;7>A78=>5F541B6? >;7:S:S:6=4:9=5S2=S:7A- ?3<S7F629D:C1=A35ADF9S;3?OI0>5J? <>5AC13G<3S@R@::5B5@:NE/4-)97@*??;@1S<(69976=BC/5;5S5=>.>9H6:7 i:9I57=54.(D5S?;6
Decrypted plaintext	Notice [1]:I need you to help me encrypt this text and hide the time, place, characters and things in it.Then you need to send the encrypted information to Mr. Zhang. Company MathWorks most complete software for computational computer produces; the main program the company that actually Passport is software MATLAB ('short for Mat rix Lab Oratory and means lab Matrix') is one of the most advanced software, algorithms and math and a programming language developed generation fourth is possible. Acknowledgements are usually put in the end of the paper, between the reference and the content. There should be no number in this part.	

Since our algorithm involves function fitting and signal decomposition, it will take more time than other classic algorithms in the encryption process. However, it can be seen from Table 1, the two versions of our algorithm can quickly complete the decryption process. Especially when dealing with long text, our proposed algorithm will be more efficient because parallel computing can be introduced after the segmentation of the plaintext in (16).

To clearly compare the difference between our algorithm of two versions, the ciphertext and secret key of the plaintext with 634 characters are listed as example in Table 2. Although there are some special characters in the plaintext, both the two versions can decrypt the ciphertext to the totally correct plaintext by using the one and only secret key.

5 Security Evaluation

5.1 Key Size

The length of the secret key has a great influence on the security of encryption. In the proposed algorithm, the length of the secret key is identical with that of the plaintext so that it will be difficult to predict. And our method does not require additional computational load when the key size increases, though the storage and transmission cost are not negligible. In the future, we will consider improving security while reducing length of the secret key.

5.2 Ciphertext or Key Attack

As the feature of Core-AFD, decryption will be successful only when the secret key and ciphertext are both completely correct. The following analysis is based on that the algorithm is known by attackers. Suppose the plaintext is text information and each character can be converted to ASCII values from 0 to 126.

In the initial version, the values of the ciphertext are all between 0 and 126 since Core-AFD is a positive frequency decomposition algorithm. Therefore, the time complexity of decryption is $\mathcal{O}(127^N)$ if intruders intercepted either the ciphertext or secret key. Similarly, the time complexity of decryption would be $\mathcal{O}(N * 127^N)$ in the enhanced version of our algorithm.

5.3 Ciphertext and Key Attack

There is another terrible scenario that both the ciphertext and the secret key are compromised. In our initial version, it may provide chance for attackers to anticipate the plaintext since the decryption process is a simple summation. With this in mind, we develop the enhanced version of our algorithm. A parameter is introduced, whose value range and position are unknown for attackers. In this situation, there is a risk of decryption only when attackers know what encryption algorithm we are adopting, but the time complexity is $\mathcal{O}(N)$. Otherwise, it is almost impossible to restore the correct plaintext.

6 Conclusion

This paper introduces the signal processing approach into the encryption and decryption. We propose a novel text encryption algorithm based on Core-AFD, which refers the idea of function fitting to convert discrete characters of the plaintext into functions to execute encryption process. The features of Core-AFD make the algorithm achieve good encryption performance. Additionally, a variable parameter is added in the enhanced version of our algorithm to improve encryption security in case the secret key and cyphertext are both captured by attackers. Experiments are carried out to verify the encryption performance and the security is evaluated at last. Actually, there is still room for improvement such as perfection of the secret key that we will present in our future work.

ACKNOWLEDGMENTS

This study is supported by the research grants: The Science and Technology Development Fund of Macao SAR FDCT 079/2016/A2, 0123/2018/A3, and MYRG 2017-00218-FST, MYRG 2018-00111-FST.

REFERENCES

[1] R. Davis (1978). The Data Encryption Standard in Perspective. IEEE Communications Society Magazine, 16(6), 5-9.

[2] G. Singh (2013). A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security. International Journal of Computer Applications, 67(19).

[3] N. Khanezaei and Z. M. Hanapi (2014). A Framework Based on RSA and AES Encryption Algorithms for Cloud Computing Services. IEEE Conference on Systems, Process and Control (pp. 58-62).

[4] A. Belazi, A. El-Latif, and S. Belghith S (2016). A Novel Image Encryption Scheme Based on Substitution-Permutation Network and Chaos. Signal Processing, 128, 155-170.

[5] T. Qian, L. M. Zhang, and Z. X. Li (2011). Algorithm of Adaptive Fourier Decomposition. IEEE Transactions on Signal Processing, 59(12), 5899-5906.

[6] T. Qian, L. H. Tan, and Y. B. Wang (2011). Adaptive Decomposition by Weighted Inner Functions: A Generalization of Fourier Serie. Journal of Fourier Analysis and Applications, 17(2), 175-190.

[7] T. Qian (2014). Cyclic AFD Algorithm for Best Rational. Mathematical Methods in the Applied Sciences, 37(6), 846-859.

[8] W. Mi and T. Qian (2012). Frequency-Domain Identification: An Algorithm Based on an Adaptive Rational Orthogonal System. Automatica, 48(6), 1154-1162.

[9] H. Akçay and B. Ninness (1999). Orthonormal Basis Functions for Modelling Continuous-Time Systems. Signal Processing, 77(3), 261-274.

[10] P. S. C. Heuberger, P. M. J. Van den Hof, and B. Wahlberg (Ed.). 2005. Modelling and identification with rational orthogonal basis functions. Springer-Verlag, London, Chapter 4. DOI: http://dx.doi.org/10.1007/1-84628-178-4.

[11] T. Qian and L. M. Zhang (2013). Mathematical Theory of Signal Analysis vs. Complex Analysis Method of Harmonic Analysis. Applied Mathematics-A Journal of Chinese Universities, 28(4), 505-530.

[12] L. M. Zhang, T. Qian, W. X. Mai, and P. Dang (2017). Adaptive Fourier Decomposition-Based Dirac Type Time-Frequency Distribution. Mathematical Methods in the Applied Sciences, 40(8), 2815-2833.

[13] S. Saitoh and Y. Sawano. 2016. Theory of reproducing kernels and applications, Springer, Singapore. DOI: http://dx.doi.org/10.1007/978-981-10-0530-5.

[14] X. Ji, T. Qian, and J. Ryan, 2000 Fourier Theory under Möbius Transformations, Clifford algebras and their applications in mathematical physics. Birkhäuser, Boston, Massachusetts, United States of America.

[15] T. Qian (2010). Intrinsic Mono - Component Decomposition of Functions: An Advance of Fourier Theory. Mathematical Methods in the Applied Sciences, 33(7), 880-891.

[16] T. Qian and Y. B. Wang (2011). Adaptive Fourier Series - A Variation of Greedy Algorithm. Advances in Computational Mathematics, 34(3), 279-293.