GOOGLY - A Cryptographic Algorithm based on Mathematical Representation of Transfer Functions of a Discrete Time

Aakanksha Vaishnavi[1], Somreeta Roy[1], Aurobinda Laha[1], Narendra Kumar Verma[1], Prof. Pranam Paul[2], Prof. (Dr) A K Bhattacharjee[3]

[1] Students, National Institute of Technology, Durgapur – 713209, West Bengal, India
[2] MCA Department, Dr. B. C. Roy Engineering College, Durgapur – 713206, West Bengal, India
[3] ECE Department , National Institute of Technology, Durgapur – 713209, West Bengal, India

Abstract:
This paper presents a new and efficient algorithm for cryptographic purpose that considers the representation of the cipher text as elements which make up the transfer function of a discrete time system in the pole-zero form. The decryption is done by conversion of the pole-zero form into the rational form and the subsequent extraction of the encrypted data from this transfer function representation. The encryption is done for normal text files. The encrypted message consists of 2 numeric arrays and a number called the 'gain'. The process of creation of a public key cipher is discussed too.

Keywords:
cryptography, algorithm, Z transform, Discrete Time System, Transfer Function, signal processing.

1. Introduction:

Cryptography is one of the most important tools that provide data and information security by hiding it. It is done through mathematical manipulation, and the format used is incomprehensible to an unauthorized person.
In this paper, a public-key cipher, termed as "GOOGLY" is presented, which is modelled for encryption of text files and representing it as the transfer function characteristics of a Discrete Time System.
Section 2 explains the process of encryption and decryption of the text read from the file specified in the code itself. Section 3 describes an implementation of the technique using the code written for the purpose and its output. It takes a real file for the execution of the algorithm. Section 4 comprises the result of using the algorithm using different files and shows a graphical study of the technique. Section 5 gives an analytical and discussion on the topic, draws a conclusion and mentions some future scopes of the projectwhich are likely to come into being.

2. The Scheme:

In mathematics and signal processing, the **Z-transform**, which is the crucial idea, used here converts, a discrete time-domain signal and a sequence of real or complex numbers, into a complex frequency-domain representation. To be more clear, in cases where a discrete-time signal $x[n]$ is defined only for $n \geq 0$, the *single-sided* or *unilateral* Z-transform is defined as

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=0}^{\infty} x[n] z^{-n}$$

In discrete-time systems, if the input to the system is x(n) and the output is y(n) the function is similarly written as H(z)=Y(z)/X(z) where Y(z), X(z) and H(z) are the Z-transforms of the output y(n) , input x(n) and impulse response h(n) of the system. When y(n) is a function of both the input x(n) and its time shifted forms as well as past outputs , then the transfer function is represented in the form of

$$H(z) = \frac{\sum_{i=0}^{P} b_i z^{-i}}{1 + \sum_{j=1}^{Q} a_j z^{-j}}$$

This is a rational form of representation. A pole zero representation resembles the form where the numerator polynomial and the denominator polynomial are both factorised to depict the poles and the zeroes of the system.

The actual description of encryption and decryption technique of a text message used during implementation of GOOGLY is presented in this section. Section 2.1 shows the ciphering technique for the text, section2.2 presents the process of key generation,, section 2.3 shows the deciphering technique of the cipher text.

2.1 Ciphering Technique:

Step 1:-To encrypt a text message at first the given text is stored in a string variable, say ***message***.
Step 2:-The length of the message is stored in a numeric variable ***len.***

Manuscript received October 5, 2009
Manuscript revised October 20, 2009

Step 3:-A one-dimensional array ***num[]*** which consists of the ASCII equivalent numbers of the constituent characters of the message is created.
Step 4:-The numbers in the even positions of the array ***num[ ]*** is extracted and stored in a numeric array ***b[ ].***
Step 5:-Similarly the numbers in the odd positions of the array ***num[ ]*** is extracted and stored in a numeric array ***a[ ]***.
Step 6:-The first element of ***a[ ]*** is stored in a numeric variable ***googly***.
Step7:-The array ***a[]*** represents the denominator coefficients of a transfer function expression of a discrete time system(DTS) and ***b[ ]*** represents the numerator coefficients of a transfer function expression of a DTS.

For example,
If b[] = [111 114 101 97 105 32 97] and
a[] = [115 109 101 116 32 115 98 100] then

$$H(z)= \frac{111 + 114\ z^{-1} + 101\ z^{-2} + 97\ z^{-3} + 105\ z^{-4} + 32\ z^{-5} + 97\ z^{-6}}{115 + 109\ z^{-1} + 101\ z^{-2} + 116\ z^{-3} + 32\ z^{-4} + 115\ z^{-5} + 98\ z^{-6} + 100\ z^{-7}}$$

where H(z) represents the transfer function of a Discrete Time System.
Step 8:- The coefficient arrays ***a[ ]*** and ***b[ ]*** are then used to compute two encrypted arrays which will be the arrays containing the zeroes ***z[ ]*** and the poles ***p[ ]*** of the above transfer function, along with a value called the gain ***k***. **The encrypted message comprises the 2 arrays *z[ ]* and *p[ ]* and the gain value *k* and the value stored in *googly*.**

2.2 Key Generation:

After the zero matrix and the pole matrix have been generated , there is a stratagem needed to reverse the steps to get back the original text. Using the zero and pole matrices, we cannot get back the exact values of the original matrices a[] and b[] since a[] and b[] are just the arrays of denominator and numerator coefficients and are expressed as part of a ratio in the transfer function. The rational transfer function deduced from the zero and pole matrices would normalize the constant term in the denominator to 1. The denominator will be of the form: $1+a_0+a_1+a_2+a_3$ and so on. So in order to maintain the integrity of the text message we need to send the constant term of the original denominator coefficient array a[] that is , the first term of a[] as the key. This key is stored as ***googly*** and transmitted to the receiver , and it is necessary to multiply the coefficient matrices deduced from the pole and zero matrices at the decryption end by this value stored in ***googly***.

2.3 Deciphering Technique:

Step 1:- ***Initially*** after getting the encrypted message the receiver will use the public key cipher ***googly*** to get back the original message. The two arrays formed with the zeroes and poles of the transfer function ie ***z[ ]*** and ***p[ ]*** and its corresponding gain value ***k*** are used to compute the arrays containing the numerator and denominator coefficients ***b[ ]*** and ***a[ ]*** respectively of the transfer function.
Step 2:- The value ***googly*** is used to multiply the arrays ***a2[ ]*** and ***b2[ ].***
Step 3:- Lastly, a final array ***num1[ ]*** is formed, whose elements would be taken alternately from the above two arrays starting from the first element of the array containing the denominator coefficients ***a[ ]*** of the transfer function. The array thus formed consists of the ASCII value of the characters in the order used in the original text message.
Step 4:- Now, the receiver can get back the original message by converting the ASCII values to their corresponding ASCII characters.We also remove the blank that we had added in the case of an odd numbered message text.

3. Implementation:

Section 3.1 illustrates the encryption of a given text in a tabular form showing the concerned matrices and Section 3.2 shows the process of decryption of the same.

3.1 Process of encryption:

Let us consider the plain text "**we wish u a merry christmas and a happy new year!**" The encryption technique is demonstrated using the Table 3.1.1

The gain value ***googly*** contains the value 32.

3.2 Process of decryption:

After receiving the cipher text in the form of the arrays z[] and p[] and the value k along with the key googly the total decryption process is shown in Table 3.2.1:

Table 3.1.1 Encryption Process

a[]	b[]	z[]	p[]
32	119	0.9671 +0.2613i	-2.1425
101	32	0.9671 - 0.2613i	0.9645+0.2494i
119	105	0.8851 +0.5153i	0.9645-0.2494i
115	104	0.8851 - 0.5153i	0.9064+0.4495i
32	117	0.7135 +0.7340i	0.9064-0.4495i
32	97	0.7135 - 0.7340i	0.6916+0.6538i
32	109	0.5267 +0.8697i	0.6916-0.6538i
101	114	0.5267 - 0.8697i	0.5418+0.8148i
114	121	0.3153 +0.9936i	0.5418-0.8148i
32	99	0.3153 - 0.9936i	0.2442+0.9540i
104	114	0.0625 +0.9839i	0.2442-0.9540i
105	115	0.0625 - 0.9839i	-0.0909+ 1.1774i
116	109	-0.2179+0.9259i	-0.0909 -1.1774i
97	115	-0.2179 -0.9259i	-0.8140 + 0.9510i
32	97	-0.4551+0.8207i	-0.8140 - 0.9510i
110	100	-0.4551-0.8207i	-0.1869 + 0.9225i
32	97	-0.5735+0.6835i	-0.1869 - 0.9225i
32	104	-0.5735 -0.6835i	-0.4987 + 0.9739i
97	112	-0.7598+0.4847i	-0.4987 - 0.9739i
112	121	-0.7598-0.4847i	-0.8226 + 0.5160i
32	110	-0.8405+0.3123i	-0.8226 - 0.5160i
101	119	-0.8405-0.3123i	-1.0579 + 0.1991i
32	121	-0.8930	-1.0579 - 0.1991i
101	97	-0.6225	-0.7686
114	33		

Table 3.2.1 Decryption Process

z[]	p[]	a[]	b[]
0.9671 +0.2613i	-2.1425	32	119
0.9671 - 0.2613i	0.9645+0.2494i	101	32
0.8851 +0.5153i	0.9645-0.2494i	119	105
0.8851 - 0.5153i	0.9064+0.4495i	115	104
0.7135 +0.7340i	0.9064-0.4495i	32	117
0.7135 - 0.7340i	0.6916+0.6538i	32	97
0.5267 +0.8697i	0.6916-0.6538i	32	109
0.5267 - 0.8697i	0.5418+0.8148i	101	114
0.3153 +0.9936i	0.5418-0.8148i	114	121
0.3153 - 0.9936i	0.2442+0.9540i	32	99
0.0625 +0.9839i	0.2442-0.9540i	104	114
0.0625 - 0.9839i	-0.0909+ 1.1774i	105	115
-0.2179+0.9259i	-0.0909 -1.1774i	116	109
-0.2179 -0.9259i	-0.8140 + 0.9510i	97	115
-0.4551+0.8207i	-0.8140 - 0.9510i	32	97
-0.4551-0.8207i	-0.1869 + 0.9225i	110	100
-0.5735+0.6835i	-0.1869 - 0.9225i	32	97
-0.5735 -0.6835i	-0.4987 + 0.9739i	32	104
-0.7598+0.4847i	-0.4987 - 0.9739i	97	112
-0.7598-0.4847i	-0.8226 + 0.5160i	112	121
-0.8405+0.3123i	-0.8226 - 0.5160i	32	110
-0.8405-0.3123i	-1.0579 + 0.1991i	101	119
-0.8930	-1.0579 - 0.1991i	32	121
-0.6225	-0.7686	101	97
		114	33

The decrypted message at the receiver end is "**wish u a merry christmas and a happy new year!**"

4. Results:

The algorithm described above works perfectly for all text formats viz. - .**txt**, **.doc**, **.rtf**, **.sys**, **.bmp**, **.exe** files.

A comparative study of the execution times for texts (during encryption) varying in length is shown in Table 4.1 below:

Table 4.1 Encryption Time

Number of letters in the text including blank spaces and special characters	Encryption time (in seconds)
1	0.017735
8	0.017987
16	0.020447
32	0.030765
40	0.032928
48	0.033877
56	0.039162
64	0.043528
72	0.046783
80	0.053458
88	0.054079
96	0.059321
100	0.065706

A graph showing the relationship between the encryption time and the text length is demonstrated next in Figure 4.1.

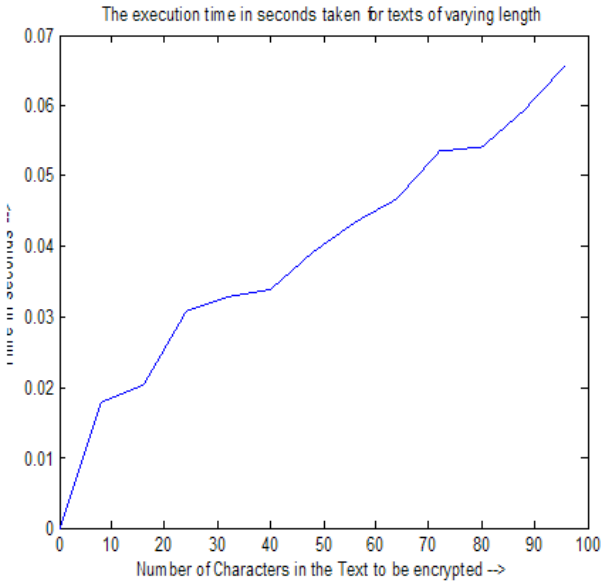

Figure 4.1 Relationship between Execution Time and Text Length

5. Analysis and Conclusion:

The advantage for this type of algorithm is that the encryption time taken is very small, in the millisecond range. However the price that we have to pay for this small execution time is that the encryption can run for 116 characters in one run. There is distortion of data seen when the text contains more characters than this. Hence there is problem in handling large volumes of data. This limitation can be removed by breaking up the large text into blocks of 116 characters and a single block which would contain a remainder number of characters when the number of characters in the text is divided by 116. Thus the program can run in blocks and the limitation is encountered.

The method described here would be useful in commercial applications where the personal information of the employees of a company have to be kept confidential or when large databases have to be rendered illegible to unauthorised users. Moreover the application of this algorithm being based on mathematical concepts used in Signals and Systems, more work can be done using such related mathematics such as Laplace transforms, Bessel functions and so on. This introduces a different dimension to the techniques used in secret key cryptography.

References

[1] J. K. Mandal, S. Dutta, "A 256-bit recursive pair parity encoder for encryption", Advances D -2004, Vol. 9 nº1, Association for the Advancement of Modelling and Simulation Techniques in Enterprises (AMSE, France), www. AMSE-Modeling.org, pp. 1-14

[2] Pranam Paul, Saurabh Dutta, "A Private-Key Storage-Efficient Ciphering Protocol for Information Communication Technology", National Seminar on Research Issues in Technical Education (RITE), March 08-09, 2006, National Institute of Technical Teachers' Training and Research, Kolkata, India

[3] Dutta S. and Mandal J. K., "A Space-Efficient Universal Encoder for Secured Transmission", International Conference on Modelling and Simulation (MS' 2000 – Egypt, Cairo, April 11-14, 2000

[4] Mandal J. K., Mal S., Dutta S., A 256 Bit Recursive Pair Parity Encoder for Encryption, accepted for publication in AMSE Journal, France, 2003

[5] Dutta S., Mal S., "A Multiplexing Triangular Encryption Technique – A move towards enhancing security in ECommerce", Proceedings of IT Conference (organized by Computer Association of Nepal), 26 and 27 January, 2002, BICC, Kathmandu

[6] William Stallings, Cryptography and Network security: Principles and practice (Second Edition), Pearson Education Asia, Sixth Indian Reprint 2002.

[7] Atul Kahate (Manager, i-flex solution limited, Pune, India), Cryptography and Network security, Tata McGraw-Hill Publishing Company Limited.

Aakanksha Vaishnavi is a student pursuing her B.Tech degree (final year) in Electronics and Communication Engineering from National Institute of Technology (NIT), Durgapur, West Bengal, India.

Somreeta Roy is a student pursuing her B.Tech degree (final year) in Electronics and Communication Engineering from National Institute of Technology (NIT), Durgapur, West Bengal, India.

Aurobinda Laha is a student pursuing his B.Tech degree (final year) in Electronics and Communication Engineering from National Institute of Technology (NIT), Durgapur, West Bengal, India

Narendra Kumar Verma is a student pursuing his B.Tech degree (final year) in Electronics and Communication Engineering from National Institute of Technology (NIT), Durgapur, West Bengal, India

Prof. Pranam Paul is associated with department of MCA of Dr. B. C. Roy Engineering College, Durgapur, West Bengal, INDIA in the. He has already submitted his Ph.D. thesis in ECE department NIT, Durgapur in the field of Cryptography and Network Security. He has total 18 research publications.

Prof. (Dr) A. K. Bhattacharjee did his Ph.D. in Engineering from Jadavpur University, Kolkata, India in 1989. Presently he is associated with Department of ECE in NIT, Durgapur, INDIA. The senior academicians, having been involved in teaching and research for last 20 years, his current areas of research are Microstrip Antenna and cryptography.