A Simple Algebraic Model based Polyalphabetic Substitution Cipher

Sukalyan Som
Department of Computer Science
Barrackpore Rastraguru
Surendranath College, Kolkata,
West Bengal, India

Mohit Kundu
Department of Computer Science
Gurudas College, Kolkata,
West Bengal, India

Sabyasachi Ghosh
Department of Computer Science
Gurudas College, Kolkata,
West Bengal, India

ABSTRACT
Cryptography is considered to be a disciple of science of achieving security by converting sensitive information to an un-interpretable form such that it cannot be interpreted by anyone except the transmitter and intended recipient. An innumerable set of cryptographic schemes persist in which each of it has its own affirmative and feeble characteristics. In this paper we have we have developed a traditional or character oriented Polyalphabetic cipher by using a simple algebraic equation. In this we made use of iteration process and introduced a key K_0 obtained by permuting the elements of a given key seed value. This key strengthens the cipher and it does not allow the cipher to be broken by the known plain text attack. The cryptanalysis performed clearly indicates that the cipher is a strong one. .

Keywords
Polyalphabetic substitution, variable length key stream, bit-ratio test, frequency test.

1. INTRODUCTION
Cryptography is considered to be a collection of tools and techniques related to components of information security such as confidentiality, authenticity, integrity, non-repudiation [1]. Cryptography refers to the science of securing data by changing the data into non-interpretable form, cryptanalysis is the science of analyzing and breaking secure communication. Cryptanalyst or attacker is a person who performs cryptanalysis [2]. Cryptology is a combination of both cryptography and cryptanalysis. A no of cryptanalytic attacks can be found wherein cipher text only, known plaintext, chosen plaintext, chosen cipher text, adaptive chosen plaintext, brute force attack, key guessing attack etc are to name a few [3].

In this paper we propose a simple algebraic model based Polyalphabetic substitution cipher wherein the plain text is converted to cipher text by the use of a key-seed value from which variable length key stream is generated producing different cipher text with every run time for similar entered plain text. In section 2 we have presented the basic terminologies. Section 3 depicts the proposed algorithm both for encryption and decryption. In section 4 we have presented the experimental results. Testing and analysis is given in section 5 comprising of frequency distribution test, bit-ratio test, encryption and decryption time comparison and comparison of plain text size with cipher file size. On the basis of testing and analysis performed conclusions are drawn in section 6.

2. FUNDAMENTAL LITERATURE
Plaintext or Cleartext refers to a message that can be understood and interpreted by the sender, the receiver and also by anyone else who gets an access to that message.

Cipher refers to the algorithm or set of algorithms for transforming an intelligible and interpretable message to unintelligible and /or un-interpretable form.

When a plain text message is codified using any suitable scheme, the resulting message is known as Ciphertext.

A key is a number or a set of numbers that the cipher, either encryption cipher or decryption cipher, as an algorithm, operates on.

The method of disguising plaintext in such a way as to hide its substance is called encryption. Decryption is the process of converting cipher-text to its original plaintext.

To encrypt a message, an encryption cipher, an encryption key and plaintext is needed which creates the ciphertext. To decrypt a message a decryption cipher, a decryption key and the cipher text is needed which reveals the original plaintext.

2.1 Symmetric & Asymmetric Cryptosystem
Depending on the key that is being used to perform encryption and decryption a cryptosystem can be classified as either a Symmetric key Cryptosystem or an Asymmetric key Cryptosystem.

Symmetric Cryptosystems use the same key, known as Secret or Private key, for both encrypting and decrypting the message. It has a problem to transport the secret key from the sender to the receiver and in tamperproof fashion.

Asymmetric Cryptosystems use one key, known as Public key, to encrypt a message and a different key, the Private Key, to decrypt it. These are also known as Public Key Cryptosystems. The private key is kept by the sender but the public key is transmitted in tamperproof fashion to the receiver.

2.2 Traditional Symmetric key Cryptography
Classical Cryptography is based on information theory appeared in 1949 with the publication of "Communication Theory of Secrecy of Systems" by C. Shannon. In Classical Cryptography both plaintext and key length were same to support secrecy through encryption [4].

Symmetric Cryptosystems can be categorized as Traditional or Character Oriented and Modern or Bit Oriented. There are two primary ways in which a plaintext can be codified to get

the ciphertext using traditional key cryptography – Substitution Ciphers and Transposition Ciphers. When these two approaches are clubbed together, we call them Product Cipher.

a. Substitution Cipher

A Substitution Cipher substitutes one symbol with another. If the symbols in the plaintext are alphabetic characters, one character is replaced with another and if the symbols are digits, one digit is replaced with another. Substitution Ciphers can be categorized as either Monoalphabetic or Polyalphabetic ciphers.

b. Transposition Cipher

A transposition cipher does not substitute one symbol for another; rather it changes the location of the symbols. Transposition ciphers may either be keyless or keyed.

In this paper we have developed a cryptosystem which uses Polyalphabetic substitution based on generated key stream we focus on Polyalphabetic substitution cipher only.

In Polyalphabetic substitution cipher, primarily proposed by Leon Battista (1568), each occurrence of a symbol may have a different substitute. To create a Polyalphabetic cipher, effort should be made to make each ciphertext character dependent on both the corresponding character and the position of the plaintext character in the message. This implies that the secrete key should be a stream of subkeys, in which each subkey depends somehow on the position of the plaintext character that uses that subkey for encipherment. In order words, we need to have a key stream k = (k_1, k_2, k_3 ...) in which k_i is used to encipher the ith character in the plaintext to create the ith character in the ciphertext.[5]

3. PROPOSED LAYOUT

3.1 Method of encryption

1. Start.
2. Input the plain text.
3. Assume a variable K and initialized with a constant value.
4. Assume a variable K_{max} and initialized with a constant value greater than K.
5. Take a character from the plain text and repeat 5 to 13 steps while the all characters of the plain text have not been taken.
6. Check the value of the variable K whether its value less than K_{max} or not.
7. If the value of K is equal or greater than K_{max} then reset the value of K with initial value.*
8. Assume one more variable N.
9. Do the following 3 steps (i.e. 10, 11, 12) for N =1 to K.
10. Convert the character to their corresponding ASCII (decimal) equivalent.
11. Add $\{5*(-1)^N+3N\}$ with the ASCII (decimal) value of the character.
12. Convert the Modified ASCII value to its equivalent character and store it as encrypted text.
13. Increment the value of K by 1.
14. Stop.

3.2 Method of decryption

1. Please Start.
2. Input the Encrypted text.
3. Initialized a variable SKP, where the value of SKP is equal to (K-1) (K variable is initialized at the time of encryption algorithm initialization.).
4. Take the a character from the encrypted text and repeat 5 to 11 steps while the all characters of the encrypted text have not been taken.
5. Convert the character to their corresponding ASCII (decimal) equivalent.
6. Add 2 to the ASCII value of the character.

Modified ASCII value = Original ASCII value of character + $\{5*(-1)^N+3N\}$
Original ASCII value of character = Modified ASCII value - $\{5*(-1)^N+3N\}$
For each character of plain text, the value of first bit of each bit Stream is = $\{5*(-1)^1+3*1\}$ (where N=1) = -5+3 = - 2
Original ASCII value of character = Modified ASCII value - $\{5*(-1)^N+3N\}$
Original ASCII value of character = Modified ASCII value - (- 2)
Original ASCII value of character = Modified ASCII value +2
Convert the Modified ASCII value to its equivalent character and store it in a New plain text.
Check the value of SKIP less than (K_{max}-1) or not.

7. If the value of SKIP is equal or greater than (K_{max}-1) then reset the value of SKIP with initial value.
8. Skip next SKP (Where SKP is a variable) character.
9. Increment the value of SKP by 1.
10. Stop.

4. Experimental results

We have implemented the above algorithms in Turbo C compiler of 16 bit by a menu driven program whose screen shots are presented for demonstration of experimental results with a sample run of the code.

Method of Encryption

Fig 1: Menu presented to the user

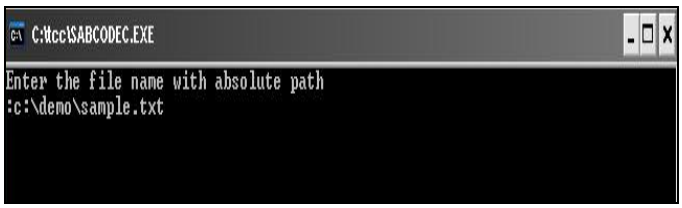

Fig 2: File name with absolute path containing plaintext

We have taken text plaintext "The quick brown fox jumps over the lazy dog." The encrypted file is presented in Fig 3

Fig 3: Plaintext file converted successfully.

The cipher text file and its contents are shown in Fig 4

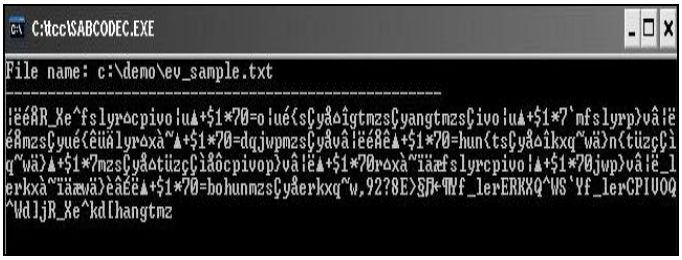

Fig 4: Cipher Text file and its contents

Method of Decryption

We use the same menu as shown in Fig 1 to provide the absolute path and file name of the cipher text file and the result after decryption is shown in Fig 5

Fig 5: Method of decryption

In the decrypted file we have recovered the plain text given "The quick brown fox jumps over the lazy dog."

5. TESTING AND ANALYSIS

5.1 The Frequency distribution test

The frequency distribution graph of source and encrypted file for the proposed algorithm is presented in Fig 6.

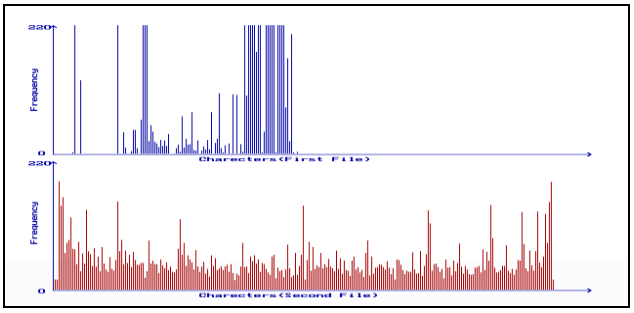

Fig 6: Frequency Distribution graph

5.3 Comparison between Original file size and Cipher file size

We have generated cipher text for various plain text file sizes from 3 bytes to 102 bytes and the cipher texts files generated are also varied of different sizes from 119 bytes to 4123 bytes which we have presented graphically in Fig 8.

It is known that if the characters in the encrypted file are evenly distributed, it will make the cryptanalysis more difficult. From the Fig it is evident that the cipher text generated by the above encryption cipher follows this criteria and thus serving the said purpose.

5.2 Bit Ratio Test

The bit ratio effect means the changes the bit values from same position between plain text and cipher text. The bit ratio can be determined as:

Bit-ratio (in %) = {(Total number of bits changed in the file after encryption) ÷ (Total number of bits present in the file)} × 100.

It has been verified that the bit ratio for our algorithm is better than existing algorithms like RSA, DES.

Table 1 is representing the average bit-ratio approximated of our proposed algorithm and a graphical presentation is also made in Fig 7.

Table 1 Bit ratio Comparison

File Name	File Size(in kb)	RSA	DES	Our Method
File01	1.80	45.40	47.57	49.2
File02	3.80	44.90	46.43	46.72
File03	8.50	45.10	47.30	47.40
File04	12.58	44.00	44.00	45.05
File05	27.00	45.10	46.80	46.23

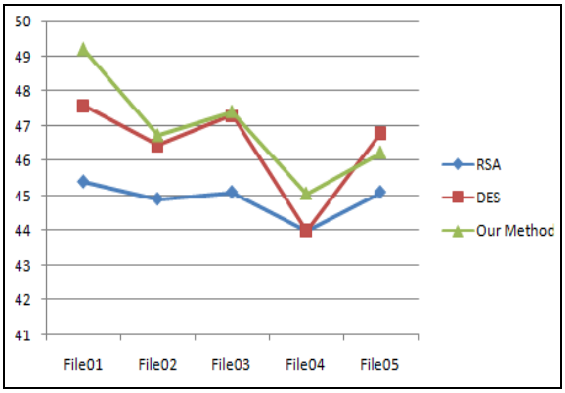

Fig 7: Bit Ratio Comparison

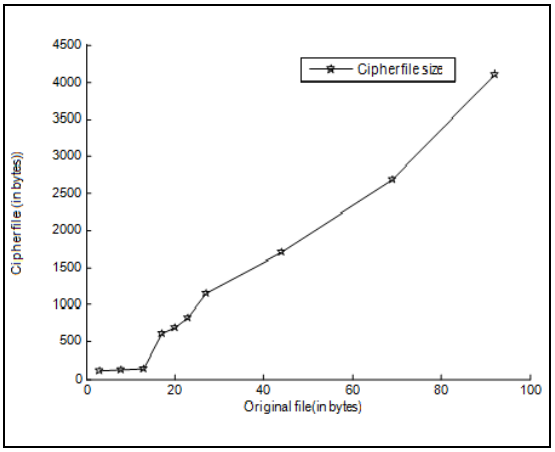

Fig 8: Original File Size and Cipher File Size

5.4 Encryption and Decryption time comparison

The time complexity indicates how efficiently the proposed algorithm will encrypt the plain text and decrypt from encrypted text. If the time complexity is much lower than time complexity on the same file of other existing algorithms, the proposed algorithm is equally good or better than existing algorithms.

Table 2 Encryption and Decryption Time comparison

Source	RSA		DES		Our Approach	
	Enc	Dec	Enc	Dec	Enc	Dec
File 01	0.45	0.31	0.12	0.67	0.4	0.32
File 02	0.5	0.33	0.1	1.1	0.48	0.35
File 03	0.32	1.01	1.01	1.1	0.35	1.05
File04	0.54	0.21	2.0	2.1	0.5	0.7
File05	0.4	1.02	3	4.0	0.3	1.0

We have presented a graphical comparison of average Encryption and average Decryption Time computed from the above table in Fig 9 from which it clearly evident that performance of our proposed work is well enough.

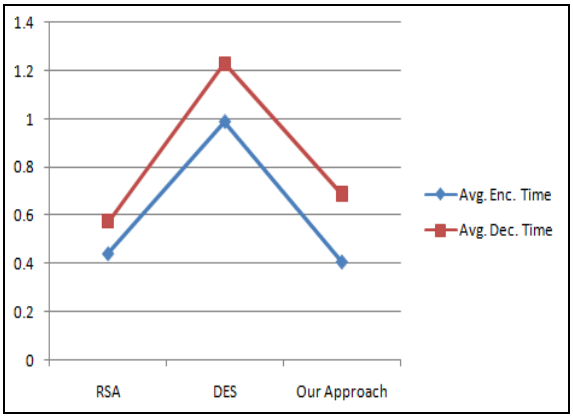

Fig 9: Encryption and Decryption Time comparison chart

6. CONCLUSION

Depending the observed experimental results, tests and analysis performed above it can be concluded that our proposed scheme is an efficient and a sufficiently strong cryptosystem providing a superior level of security. From the comparison based on Encryption and Decryption time, it has been well proved that with regard to time complexity, use of our proposed scheme is always advantageous. The comparison of Frequency Distribution also depicts that the encrypted characters i.e. cipher texts are evenly distributed and it has been made more difficult for the cryptanalysts to recover plain text from cipher text. Bit-ratio test also proved to be equally good with existing well-known ciphers like RSA and DES.

To conclude the proposed algorithm is a simple, straightforward and compact approach to develop a cryptosystem using the essence of elementary algebraic operations which provides the equivalent or sometimes even better level of security using optimally minimal time complexity. A comparative study and security level will be verified in future with other well known classical algorithms like for example Vigenere cipher and even its further modifications.

7. REFERENCES

[1] Bement A. L. et. al. (2004), Standards for Security Categorization of Federal Information and Information Systems, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg, MD 20899-8900.

[2] Ayushi, (2010), A Symmetric Key Cryptographic Algorithm, International Journal of Computer Applications (0975 - 8887) Volume 1. No. 15.

[3] Atul Kahate, (2008) Cryptography and Network Security, Tata McGraw-Hill Education, pg. 47.

[4] Ijaz Ali Shoukat , Kamalrulnizam Abu Bakar and Mohsin Iftikhar, "A Survey about the Latest Trends and Research Issues of Cryptographic Elements", p 141, International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, May 2011, ISSN 1694 0814.

[5] Sukalyan Som, Saikat Ghosh, "A Survey of Traditional or Character Oriented Symmetric Key Cryptography", International Journal of Advanced Research in Computer Science, Vol. 2, No. 4, July-August 2011

[6] R. Venkateswaran, Dr. V. Sundaram, "Information Security: Text Encryption and Decryption with Poly Substitution Method and Combining the Features of Cryptography", p28-30, International Journal of Computer Applications, Vol. 3, No. 7, June 2010.