Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 10; April-June, 2015 pp. 92-94
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/acsit.html

T's Random Cipher Algorithm

Tushar K Koshti[1] and Anamika K Mittal[2]

[1,2]GTU Affiliated
E-mail: [1]koshtitushar1994@gmail.com, [2]anamika5285@gmail.com

Abstract—*Information Security is a major challenging issue in today's technological world. There is a huge demand for a stronger and secure encryption techniques which is very hard to crack. Earlier so many researcher have proposed various encryption technique and algorithm such as DES, AES, Blowfish, RSA, Feistel block cipher etc. Some of them are very popular in achieving data security at a great extent like Blowfish and AES. But nowadays as security level is increased the time span and complexity of algorithm to perform various operations is also increased. This is the major cause of decreasing efficiency and speed of an encryption techniques. For this we have proposed a new encryption technique named "T's Random Cipher Algorithm (TRCA)" with a random key and random cipher generation which enhance security level as well as speed and efficiency of an encryption system. The (TRCA) is a stream cipher algorithm which is applied on a plaintext with a random key generation where the key is randomly generated every time we encrypt the plaintext. In this abstract we have proposed a new invented encryption technique which is very secure and very efficient. (TRCA) is implemented in a way such that each time the cipher text generated is different from the previously generated cipher text even for the same plaintext. So because of this it is difficult or even impossible to decrypt the message.*

1. INTRODUCTION

With increase in the use of internet there has been an increased opportunity in identity fraud, organized crime and various other forms of cybercrime.

2. MATHEMATICAL METHODS AND OPERATIONS INVOLVED

Module1: Key selection, distribution and sub key generation algorithm.

Module2:Encrypt the rich text usingPoly-alphabetic cipher.

Module3: Perform 16 rounds of transpositions on the output of module2 based on the keys generated in module1.

The output of this stage is the final cipher intended for the secure transmission.

Note: Reverse of this process is called the decryption

The brief explanation and the methods used in each module are explained below:

2.1 Key Selection & Distribution

The key is selected by Random Number Generation technique, in which every time a new random key is obtained to apply XOR function on the plaintext because of which the key obtained every time is unique.

2.2 Encryption using Poly-alphabetic cipher

In this step, the cipher of every same alphabet of the plaintext is produced is unique. The key which is applied on the plaintext keeps on changing every time the next byte of the plaintext is taken to cipher.

2.3 Perform 16 Rounds of Transpositions:

The cipher which is produced by the polyalphabetic cipher is taken as input in this step. The cipher is again ciphered by 16 rounds of XOR function.

3. PROCESS OF ENCRYPTION AND DECRYPTION -

3.1 Encryption

1. Read plain text.
2. Get Cipher text by applying polyalphabetic cipher technique.
3. Convert cipher text into blocks.
4. Apply Feistel cipher for 16 rounds.

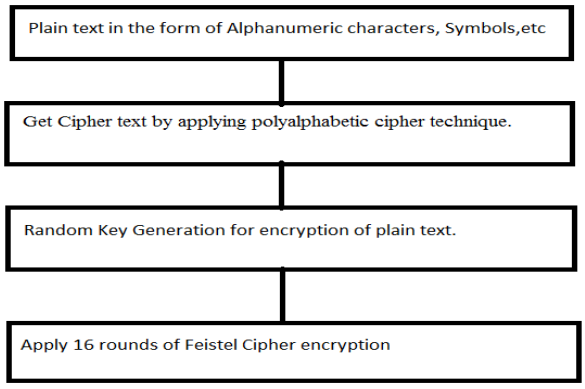

Fig. 1: Encryption process

3.2 Decryption

1. Receiver receives encrypted key.
2. Decrypts it by Feistel cipher decoding.
3. Decrypts it by polyalphabetic cipher decoding.

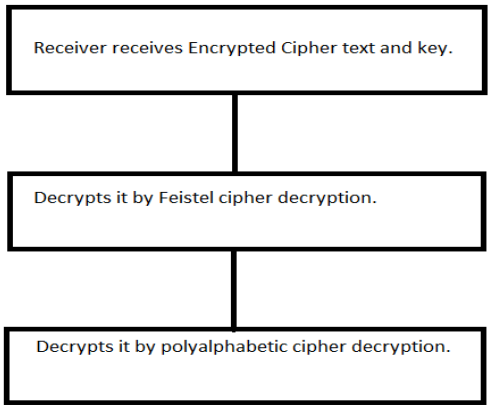

Fig. 2: Decryption process

4. DEVELOPMENT OF CIPHER

In this we have considered a plain in the form of alphanumeric characters, symbols, etc. For the development of the cipher we have several phases in this algorithm as shown in the Fig. 1.

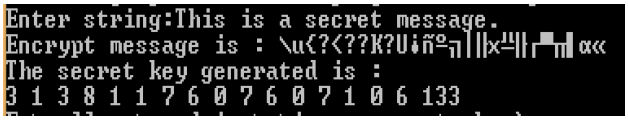

Fig. 3: Plain text 1 considered for encryption

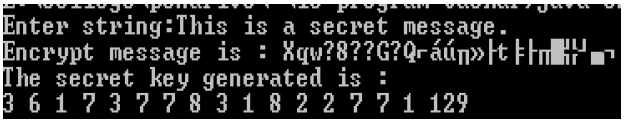

Fig. 4: Same plaintext but different cipher

To exhibit and prove a strong avalanche effect we have considered plain text in which we have not changed any characters in Fig. 3 & 4. As shown in Fig. 3 & 4, Even if the plaintext is same every time the plain text is encrypted is different from the previous cipher.

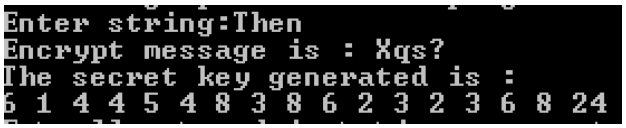

Fig. 5: Plain text considered for encryption

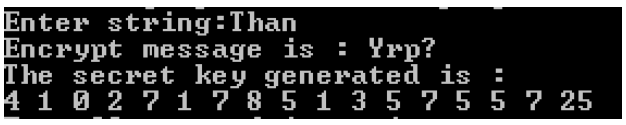

Fig. 6: Only one character of plaintext is different from previous example

A desirable property of any encryption algorithm is that a small change in either the key or the plain text should produce a significant change in the cipher. In particular, a change in one bit / character of the plain text or key should produce a change in many bits / characters of the cipher text. The same we have proven in this algorithm. The resultant output after changing a single character in the first input is as shown in the Fig. 5 & 6. In Fig. 5 & 6, the cipher text and key is changed by great difference even if the change is plain text is of one character only.

5. CRYPTANALYSIS

The cryptanalyst assails which are usually measured in the literature of Cryptography are:

1. Cipher text only attack (Brute force attack)
2. Known plaintext attack
3. Chosen plaintext attack
4. Chosen cipher text attack

In this investigation the key = K' is a 32 digits alphanumeric character. From the Left hand side first 22 characters (15 + 7) are used to compute the starting address and increment value for color substitution, next character (23^{rd}) is used to select integral function and tree traversal method, and the remaining 9 characters (24^{th} to 32positions) are used as a key for permutation. In this we have three possibilities:--

Case 1: key can be only characters: Because, the alphabets are only 26, to enter 32 characters in the key, obviously some characters will be repeated. In these circumstances:

Maximum number of Keys = $(26)^{32} = 1.9 \times 10^{45}$ Keys

If the time required for resolving of the plain text for one value of the key in the key space is taken as 10^{-3} seconds, then time required for obtaining the plain text by considering all the possible keys in the key space is $1.9X10^{45}X10^{-3}$ if we perform one encryption per micro second it takes

$$\frac{1.9 \times 10^{45} \times 10^{-3}}{365 \times 24 \times 60 \times 60} = 6 \times 10^{35} \text{ Years}$$

Case 2: Out of 32 characters, first 26 can be characters and the remaining 6 can be numbers between 0 to 9. In this situation:

Maximum number of Keys = $(26)^{26} + (10)^6 = 6 \times 10^{36}$ Keys.

If we perform one encryption per micro second it takes

$$\frac{6 \times 10^{36} \times 10^{-3}}{365 \times 24 \times 60 \times 60} = 1.9 \times 10^{27} \text{ Years}$$

Case 3: key can be only numbers: Because the

key length is 32 and the numbers can be any decimal number between 0 to 9‘, naturally the numbers will be repeated in the key. In this condition:

Maximum number of keys= $(10)^{32.}$

If we perform one encryption per microsecond it takes:

$$\frac{10^{32} \times 10^{-3}}{365 \times 24 \times 60 \times 60} = 3.1 \times 10^{29} \text{ Years}$$

In all three cases the number of possible keys was large, and the time required to try all probable keys is too high. Brute force attack is not possible and hence; it is impossible to break the cipher.

In the case of known plain text attack, we have to know as many pairs of plaintext and cipher text as we require. The combination of plain text and cipher text changing even for slightest change in plaintext and even if the plain text is same the cipher text produced will be different from the previous one. So having of combinations of plain text and cipher text is of no use.

With permutations and substitutions in different stages we can conclude that knowing plain text does not work. In the last two cases of the cryptanalysis attack, no scope is found for breaking the cipher. Other then all these, to prove that the cipher is potential one, it is mandatory that the cipher should confirm a strong avalanche. To reveal and confirm a strong avalanche effect we have considered one more plain text in which we have changed a single character in the first plain text as explained in section 4 and it is shown figure.

It is noticeable that the only third character in the plain text is differed in Fig. 5 & 6. We have also encrypted the new plain text with the same procedure and experimentally more than 90% of the cipher in the second experiment is differ from the first experiment. In view of the above conversation, we conclude that the Cipher is a potential one.

6. RESULTS

The T’s Random Cipher Algorithm confirms that it comfortably converting all kinds of text, symbols. The greatness of this algorithm is that it works in 16 rounds, and supports the stream cipher. In this, we have used two types of permutations and it’s a new dimension in the cryptography. The strength of any algorithm depends on key rather than the algorithm, in this the length of the key goes on changing according to the plain text and proven that it is far from crypt analysis attacks and especially it gives a strong avalanche effect.

7. CONCLUSION

In this paper we have developed T’s Random Cipher algorithm i.e. a symmetric stream cipher generation algorithm using multiple transformation. In this we have used two types of transposition algorithms to improve the complexity and have 16 rounds of permutations. We have proven that it can encrypt / decrypt all kinds of text, numbersand symbolswith example as shown in figures. For performing one encryption per micro second it takes minimum 1.9×10^{27} years.

For transferring key from sender to receiver we have used an enhanced RSA algorithm. Especially we have concentrated on the sub key generation algorithm, explained the three possible cases and its time complexity.

Lastly, we conclude that, with the alphanumeric key, the cipher is very strong and the algorithm is potential one.

REFERENCES

[1] Denning, D., F. Ayoub,―Cryptographic techniques and network security‖, IEEE proceedings, Vol 131, 684694,Dec 1984.

[2] Stalling,―Cryptography and network security‖, Fourth edition, LPE, 81-7758-774-9.

[3] Ravindra babu, Udayakumar and Vinaya babu ‖ A Survey on Cryptography and Steganography Methods for Infromation Security‖, IJCA, 09758887, Vol 12, No-2, Nov2010.

[4] Ravindra, Udaya and Vinaya babu, An Improved Playfair Cipher Cryptographic Substitution Algorithm, JARCS, (0976-5697), Volume 2, No-1, Jan-Feb 2011.

[5] Ravindra, Udaya and Vinaya babu, An Extension to traditional Playfair Cipher Cryptographic Substitution Method, IJCA, (0975 – 8887), Volume 17, No-5, March 2011.

[6] Ravindra Babu, Udaya Kumar and Vinaya babu, An Enhanced Poly alphabetic Cipher using Extended Vigenere Table, IJARCS, (0976-5697), Volume 2, No.2, Mar-April 2011.

[7] Ravindra Babu, Udaya Kumar and Vinaya babu, An Enhanced and Efficient Cryptographic Substitution Method for Information Security, IJMA, Archive-2 (10), 2078-2083, Oct 2011.

[8] Ravindra, Udaya Kumar and Vinaya babu, A Contemporary Poly alphabetic Cipher using Comprehensive Vigenere Table, WCSIT,(2221-0741), Vol.1,No 4, 167-171,2011

[9] Alaa, Bilal, A fast approach for braking RSA cryptosystem, WCSIT,2221-0741,Vol 1,No 6, 260263, 2011.

[10] Ravindra Babu, Udaya Kumar and Vinaya babu, An Enhanced RSA public key cryptographic algorithm, communicated to IJARCS, 0976-5697, Vol-2, No 5, 497499, Sep-Oct 2011. .

[11] Yedidyah, Moshe J. Augenstein, M.Tenebaum, Data Structures Using C and C++, 2nd Edition, 249319, ISBN-81-203-1177-9, Jan 2000.

[12] Ashok N. Kamthane, C and Data Structures, Pearson Education, ISBN 81-297-0261-4, pages: 572-576, First edition, 2004.

[13]