An Encryption Algorithm Functioning on ASCII Values and Random Number Generation

M. Lavanya¹*, R. Vijay Sai¹, A. Festina², J. Eshwari², T. Manopriya² and V. Vaithiyanathan¹

¹School of Computing, SASTRA University, Thanjavur – 613401, Tamil Nadu, India; m_lavanyass@ict.sastra.edu, vvn@it.sastra.edu
²Information and Communication Technology, SASTRA University, Thanjavur – 613401, Tamil Nadu, India; vijaysai@it.sastra.edu, festinaantonyraj@gmail.com, eshwarijeevan@gmail.com

Abstract

This article aims to focus on encryption based algorithm on ASCII value and Random number generator. Proposed algorithm has been developed with the view to achieve data security and prevent unauthorized persons from meddling with such secret data. An ASCII based conversion has been implemented to encrypt shorter length messages through a symmetric key. The algorithm specifically accommodates the concept of random number generation that makes it tough for the hackers to decrypt despite accidentally discovering the key. This concept also involves floating point numbers for encryption to enhance the difficulty level of cryptanalysis. Hence this algorithm will provide an efficient way to make it almost impossible for the intruders to track the message.

Keywords: ASCII, Encryption, Hackers, Random Number, Symmetric Key

1. Introduction

“Cryptography is the essential building block of independence for organizations on the internet just like armies are the essential building blocks of states, otherwise one state just overtakes another” says Julian Assange. In the existing substitution ciphers such as Vignere cipher, Vernam cipher and Caesar cipher one of the main problems is that they are so vulnerable to frequency analysis. They are highly prone to attacks because the cipher text can easily be broken by mapping the frequency of its letters. To overcome these drawbacks, this paper proposes the RAND_ASC encryption algorithm which cannot be attacked by brute force attack and provides high data security. The target of this algorithm is to provide high data security for the text sent via emails on internet and defies any attempt of code breaking by frequency analysis.

Two important characteristics of cryptography¹ are the strength of the encryption algorithm and the secrecy of the key. A good number of papers have been proposed with this idea targeting a goal of preserving the secret text from brute force attack with ASCII code which is significant in computer systems.

In paper² proposes a new encryption algorithm based on modulus and remainder operations on ASCII code and a substitution array approach but the setback of the paper is that the number of keys used is equal to the number of characters in the plain text. In paper¹ put forward a symmetric encryption algorithm where a randomly generated key is converted to another string based on the ASCII value of the plain text characters and is then used...
An Encryption Algorithm Functioning on ASCII Values and Random Number Generation

2. Proposed Technique

This paper proposes a new encryption algorithm called RAND_ASC encryption algorithm which performs substitution cipher based on the American Standard Code for Information Interchange (ASCII), a character encoding scheme. The algorithm uses a symmetric key which is a floating point number for encryption. The prime concept is to convert the plain text into a format of floating point numbers. To make cryptanalysis arduous, random numbers are generated which are coded with the cipher text. The use of floating point numbers repudiates the mapping of frequency of characters and hence brute force attack is impossible and the encrypted text is indecipherable.

3. Encryption Algorithm

PSEUDOCODE: RAND_ASC ENCRYPTION ALGORITHM
BEGIN
INPUT: Plain text (PT), Key (K)
1. Choose the plain text (PT) to be encrypted  
2. n = LENGTH(PT)  
   // n = Number of characters in plain text
3. FOR (i = 1 upto n)  
4. // Convert each character to its ASCII value T_i = ASCII(PT_i)  
   // T = Temporary text
5. // Extract int part of key and sum it T_i += EXTRACT_INTPART(K)  
6. // Find the prime number corresponding to index i  
   T_i = MUL(T_i, GET_PRIME)  
7. // Calculate the square root T_i = SQUARE_ROOT(T_i)  
8. // Multiply the key with T_iT_i = T_i * K  
9. // Assign T_i to Cipher text C_T_i  
   C_T_i = T_i
10. // Generate random numbers R1, R2...Rb and concatenate with C_T_i
END LOOP
END

OUTPUT: Cipher text (CT)

4. Decryption Algorithm

PSEUDOCODE: RAND_ASC DECRIPTION ALGORITHM
BEGIN
INPUT: Cipher Text (CT), Key (K)
1. Choose the cipher text to be decrypted  
2. n = LENGTH(CT)  
3. FOR (i = 1 to n)  
4. FOR (EXTRACTED FPN)  
   // Floating point number
5. IF( NOT RAND)  
6. T_i = C_T_i  
   // T = Temporary text
7. // Divide T_i by Key T_i = T_i / K  
8. // Calculate the square of T_iT_i = SQUARE(T_i)  
9. // Find the prime number with respect to i  
   T_i = DIV(T_i, GET_PRIME)
10. // Extract int part of key and subtract $T_i = T_i - (\text{EXTRACT\_INTPART}(K))$
11. // Round off $T_i$ value $T_i = \text{ROUND}(T_i)$
12. // Find the character by ASCII value $PT_i = \text{CHAR}(T_i)$
13. END IF
14. END LOOP
15. ENDLOOP

OUTPUT : Plain text (PT)

5. Observations from the Analysis

- RAND_ASC algorithm completely defies brute force attack and the frequency of the characters can never be mapped because of the uniqueness of floating point numbers coded with the generated random numbers.
- Since the length of plain text is different from that of the cipher text the chances of breaking the code is NIL.

6. Advantages of using RAND_ASC Encryption Algorithm

- The Length of Key (K) is variable.
- The Relative Frequency (RF) of characters can never be mapped because this algorithm does not reflect frequency data of the original alphabet.
- Brute force attack is impossible because the Key (K) is a floating point number. So the range and accuracy of key is unviable to predict.
- ASCII based conversion proves to be a great advantage because the printable ASCII value of all characters lies within a range of (0 – 127). Hence only accuracy determines successful decryption which shields the cipher text from attacks.
- Random numbers coded along with cipher text conceals the secret data. The level of secrecy is high.
- All numbers, alphabets, special characters and sentences can be encrypted with RAND_ASC encryption algorithm.
- This symmetric key encryption technique is relatively fast and the execution time is less.

Based on the comparison and analysis of RAND_ASC algorithm with other encryption algorithms is shown in Table 1 and also in Figure 1. Absence of Relative Frequency (RF) experimental values were shown in Figure 2.

<table>
<thead>
<tr>
<th>Table 1. Comparison and analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric key</td>
</tr>
<tr>
<td>Brute force attack</td>
</tr>
<tr>
<td>Absence of relative frequency</td>
</tr>
<tr>
<td>Random number Generation</td>
</tr>
<tr>
<td>PT.length = CT.length</td>
</tr>
<tr>
<td>PT.length = Key.length</td>
</tr>
</tbody>
</table>

Figure 1. Comparison chart.
7. Conclusion

Cryptography is significant for the secured transmission of data on internet. The RAND_ASC encryption algorithm that has been formulated incorporates confidentiality, integrity and availability and ensures high data security, enabling the protection of text sent via emails. It defies brute force attack and relative frequency analysis. Hence this algorithm can be implemented in real time applications. Hackers will find it almost impossible to intervene and take possession of official records or pilfer credit card data and make access to others accounts in financial sectors and so this will be very efficient and useful to the Government, other institutions, as well as the public at large.

8. References