Pseudorandom Numbers Generation: An Implementation To A Secure Cryptosystem

Sravani Jayanti[1], K Chittibabu[2], Chandra Sekhar Akkapeddi*[3],

[1,2]Research Scholar, Dept. of Mathematics, GITAM, Visakhapatnam, 530045, India,
[1]Email: sjayanti@gitam.in, [2]E-mail: 121962101201@gitam.in
*[3]Professor, Dept. of Mathematics, GITAM, Visakhapatnam, 530045, India, E-mail: cakkaped@gitam.edu
***Corresponding Author:** Prof. Chandra Sekhar Akkapeddi

Abstract

The sharing of information over secure channels relies on the efficiency of the cipher. In particular, designing mathematical models using programming techniques play a vital role in developing a secure cryptosystem. In this paper, we developed a pseudorandom numbers generator using modular arithmetic for performing encryption in a stream cipher which applies Affine operation.

DOI Number: 10.14704/nq.2022.20.9.NQ440104 **Neuro Quantology 2022; 20(9):944-947**

Introduction

In the era of digitalization, information shared over secure channel should retain confidentiality. The subject of cryptology plays a crucial part in exchanging the information securely. Cryptology is the combination of cryptography and cryptanalysis. Cryptography is the art of encrypting and decrypting data by means of a shared key (public and/or private). Cryptanalysis are developed. Some of the famous modern cryptosystems are AES and DES.

The information to be shared is stored in the form of bits. The allocated data is either communicated in few blocks of bits or one bit at a time depending upon which ciphers are divided as Block or Stream cipher. Block ciphers encrypt group of bits at a time and retain the confusion and diffusion property and hence is the art of breaking the cryptosystems to hack enhance the security. On the other hand, Stream the information [1-2]. An efficient cryptosystem, a secure key exchange protocol and a robust key generation technique play a vital role in communicating data among people. These algorithms are developed using different mathematical concepts and are implemented using programming languages. Further, the algorithms are tested against various parametersfor measuring their efficiency. Few of themeasuring parameters are time, memory, andcomplexity of an algorithm, randomness of the key generated, speed of the key and cryptosystems' resistance to different attacks. In the ancient era, classical ciphers such as Shift ciphers and multiplicative ciphers were designed. With the onset of technology, therewas a substantial need for efficient cryptosystems whose efficiency should outreach the measuring parameters discussed above. Thus, modern cryptosystems along with key exchange protocols ciphers encrypt one bit at a time, retain confusion property only and are desirable to be used as one time pads by using pseudorandom numbers.

One Time Pad and Pseudorandom numbers generator:

One Time Pad is an attack-resistant encryption technique which uses a pre-shared random key stream. The length of the plaintext and that of the key stream is same. The random key stream is produced using a pseudorandom numbers generator which intakes seed value to generate numbers. The seed value can be input either manually or it can be recorded from the hardware parameters such as number of mouse clicks in a certain period of time. Linear recurrence relations are substantially used to generate pseudorandom numbers which include Fibonacci series, Lucas numbers and many more [3-7].

Classical ciphers are applied to develop strong cryptosystems by overcoming their drawbacks [8]. In this paper, a stream cipher which uses affine cipher for performing encryption using a random key stream generated using modular exponentiation and modulo addition is developed.

Affine Cipher:

An Affine cipher is mono-alphabetic cipher which uses two private keys to encrypt the data. For a total of n number of characters, the private keys are selected as: $a,\ b \in N \ni \gcd(a, n) = 1$, $a,\ b < n$. The plaintext P is encrypted as: Cipher text, $C = (a * P + b)\ mod\ n$ and the cipher text C is decrypted as: $P = (C - b) * a^{-1}\ mod\ n$ where $a * a^{-1} \equiv 1(mod\ n)$. This cipher is vulnerable to attacks by means of frequency analysis [1-2].

Proposed Method

The proposed cryptosystem inputs the length of the plain text and the exchanged key to generate a sequence of random numbers using which each character is encrypted by using the encryption technique of an Affine cipher.
The Methodological approach for the proposed cryptosystem is provided in the form of a pseudo code.

The parameters used in the pseudo code represent the following:
$m = Total\ number\ ooo\ characters$
$P[ii] = Plaiin\ text\ at\ ii^{th}\ posiitiion$
$C = Ciipher\ text\ at\ ii^{th}\ posiitiion$
$n = length\ ooo\ the\ plaiin\ text/ciipher\ text$
$Shared\ Key, K = (p, a, b) where\ p \geq m, p \geq n\ and\ p\ iis\ priime, a, b \in N$
$a[\ ] =$
$Sequence\ ooo\ random\ numbers\ generated$

Key Generation (Pseudo code to generate random numbers):
Start
$a[1] = a\ mod\ p$
$a[2] = b\ mod\ p$
$ooor\ ii = 3\ to\ n$
{
Start
$ooor\ ii = 1\ to\ n$
{
$C[ii] = (P[ii] * a[ii] + a)\ mod\ p$
}
End

Pseudo code to perform decryption:

Start
$ooor\ ii = 1\ to\ n$
{
$P[i] = ((C[i] - a) * a[i]^{-1})\ mod\ p \quad //where, a[i] * a[i]^{-1} \equiv 1(mod\ p)$
}
End

Implementation

The proposed method is implemented for a character set consisting of 26 alphabets whose numerical equivalents are used in computations. The 26 alphabets are mapped to the numerical values as: $A \leftrightarrow 1, B \leftrightarrow 2, \ldots, Z \leftrightarrow 26$.
For the plain text "SECRET" and the shared key combination (29,15,19), the cryptosystem designed generates the following result.

$m = 26, p = 29, a = 15, b = 19, n = 6, Plain\ text = SECRET \leftrightarrow 19\ 5\ 3\ 18\ 5\ 20$						
Key Stream	$a[1] = 15$	$a[2] = 19$	$a[3] = 13$	$a[4] = 10$	$a[5] = 2$	$a[6] = 15$
Cipher text	$C[1] = 10$	$C[2] = 23$	$C[3] = 25$	$C[4] = 21$	$C[5] = 25$	$C[6] = 25$
Plain text	$P[1] = 19$	$P[2] = 5$	$P[3] = 3$	$P[4] = 18$	$P[5] = 1$	$P[6] = 20$

2

Result Analysis

```
#include<iostream>
#include<cmath>
using namespace std;
long int ec(int m, int n, int k)
{
    long int m1=1;

    for(int i=1;i<=n;i++)
    {
        m1=m*m1;
        if(m1>=k)
        {
            m1=m1%k;
        }
    }
    return m1;
}
```

1. The Pseudo random numbers generators algorithm is implemented using DEV C++ compiler. The following code is used to generate a pseudo random sequence ofnumbers:

}
End

$a[i] = (a[i-1]^{a[i-2]} + i) \bmod p$

Pseudo code to perform encryption:

that the sequence of numbers would repeat after certain values which is advantageous to the intruder for conducting chosen plain text attack.

```
int main()
{
    long int a,b,p,n;
    long int a1[1000],b1[1000],d1[1000];
    cout<<"Enter a value:";
    cin>>a;
    cout<<"Enter b value:";
    cin>>b;
    cout<<"Enter p value:";
    cin>>p;
    cout<<"Enter n value:";
    cin>>n;
    a1[0]=a%p;
    a1[1]=b%p;
    for(long int i=2;i<n;i++)
    {
        b1[i]=ec(a1[i-1],a1[i-2],p);
        d1[i]=(b1[i]+i+1)%p;
        a1[i]=d1[i];
    }
    for(int i=0;i<n;i++)
    {
        cout<<a1[i]<<"\t";
    }
    return 0;
}
```

After executing the code against different prime numbers (less than 100) and 5 different key combinations (p, a, b), we obtain distinct pseudorandom sequences of numbers which is periodic after certain values. The following data is formulated after execution:

Prime, p	Periodic after certain random values	Period
$p = 2$	3	4
$p = 3$	3	3
$p = 5$	5 or 10	5
$p = 7$	7 or 14	14
$p = 11$	9 or 0 or 13	33
$p = 13$	7 or 16	13 or 39
$p = 17$	44	17
$p = 19$	7	95
$p = 23$	1 or 83	23
$p = 29$	53 or 82	29
$p = 31$	63 or 32	62
$p = 37$	12 or 49	37
$p = 41$	86	41
$p = 43$	135 or 50	43
$p = 47$	46	235
$p = 53$	97	53
$p = 59$	101	118
$p = 61$	44	182
$p = 67$	226	402
$p = 71$	65	710
$p = 73$	27	146
$p = 79$	254	158
$p = 83$	671	166
$p = 89$	174	356
$p = 97$	108	291

From the data, the periodicity of the obtained sequence depends upon the prime number $'p'$ and is given by the relation $t = k * p$, $k \in N$, where t is the period of the pseudorandom sequence generated. Hence, while performing encryption the condition $p \geq n$ shall not be dropped because for $p < n$ there is a possibility

2. The developed cryptosystem uses affine ciphers' algorithm for encryption but it is not prone to attacks by means of frequency analysis. In the example discussed above, we can observe that the character "E" occurs twice in the plain text but the cipher text for the character "E" is different in both cases. Also, in the cipher text we have three character values to be "25" but their corresponding plain text characters are different. Thus, the cryptosystem cannot be broken down by means of frequency analysis. This is because each character is encrypted using a different key value. The key generation algorithm helps to generate a random key stream which is used to encryptthe data.

Conclusion

The developed cryptosystem is a stream cipher which encrypts one character at a time. The pseudorandom numbers generator produces a sequence of numbers which is periodic after and that of the key stream is same and hence it is desirable to use the encryption algorithm as a One Time Pad. The proposed method is an application to the Affine cipher. The security of the cryptosystem proposed is dependent upon the key exchanged which generates a pseudorandom key stream which can be enhanced by means of a strong and an efficient key exchange protocol.

Acknowledgment

This work is supported by GITAM in the form of Dr. M.V.V.S. Murthi Research Fellowship for which we are grateful.

References

Forouzan, B.R. and Debdeep, M. 'Cryptography and Network Security', Third Edition, McGraw Hill Education.

Stinson, D.R. and Paterson, M.B. 'Cryptography Theory and Practice', Fourth Edition, CRC Press, Taylor & Francis Group.

Amiruddin, A. *et al.* (2019) 'Construction and Analysis of Key Generation Algorithms Based on Modified Fibonacci and Scrambling Factors for Privacy Preservation', *International Journal of Network Security*, 21(2), pp. 250-258. Available at: (DOI: 10.6633/IJNS.201903_21 (2).09)

Chandra Sekhar, A. *et al.*(2007) 'Data Encryption technique using Random number generator', 2007 *IEEE International Conference on Granular Computing,* pp. 576-579, Available at: DOI 10.1109/GrC.2007.73

Sudha K.R. *et al.*(2007) 'Cryptography protection of digital signals using some Recurrence relations', *International Journal of Computer Science and Network Security*, 7(5), pp. 203-207. Available at: http://paper.ijcsns.org/07_book/200705/20070530.pdf

Behrouz, F.V. and Rahim, A. (2016) 'A Novel Pseudo-Random Number Generator for Cryptographic Applications', *Indian Journal of Science and Technology,* 9(6), pp. Available at: DOI: 10.17485/ijst/2016/v9i6/73922.

David, D.C. (2012) 'Random Number Generation: Types and Techniques', A Senior Thesis submitted in partial fulfillment of the requirements for graduation in the Honors Program Liberty University. Available at: https://core.ac.uk/download/pdf/58824567.pdf,

Ofelius, L. *et al.* (2019) 'Application of Linear Congruent Generator in Affine Cipher Algorithm to Produce Dynamic Encryption', *1st International Conference of SNIKOM 2018, Journal of Physics: Conference Series,* 1361 012001, pp. 1-6. Available at: doi:10.1088/1742-6596/1361/1/012001.