Algorithm for Text Data Encryption by Position Swapping based on LFSR Pseudorandom Key Generation

Nitin Kaul
Electronics and Communication Engineering
Department, Lovely Professional University
Punjab, India

Shikha
Computer Science Engineering Department, Lovely
Professional University
Punjab, India

ABSTRACT

Data security is one of the main issues as the technology is rising. Daily infinite amount of data is being transmitted over the internet, through mobile phones, laptops, and it is of more concern that, that data should be secured. Cryptography gives us number of ways by which the security can be enhanced. Number of encryption algorithms and protocols are there, by which the data can be made more and more secure. Cryptography is the science where security engineering meets mathematics. In this paper, a basic two level encryption algorithm has been proposed based on the position swapping of data values according the values of key and LFSR pseudorandom sequence generation for making the transmission of text data secure.

Keywords

Cryptography, LFSR pseudorandom key, Encryption, Decryption, Text data

1. INTRODUCTION

Digitalization has taken the technology to new heights. Whether it is about the reservations, banking, shopping, everything is now online. With digitalization, one problem arises and that is the security of the data being uploaded. Privacy of the information is of major concern when the issue is about online data uploading. That is why cryptography plays an important role for the security of data of the user. Cryptography word came from Greek word '*kruptos*' which means 'hidden'. And so this science is called as the science of security [5]. Cryptography is the science which deals with the encryption of data to make it more secure from the eavesdropper and then transmit it to the receiver. Encryption means to change the data such that the eavesdropper will not be able to find the original data. And the main component of a perfect encryption is the key by which the data is encrypted. Both the encryption and decryption uses this key, means they both are dependent on this key.

$$E_K\,(M) = C$$

$$D_K\,(C) = M$$

There are some algorithms which use different encryption and decryption key. That is the reason the key management in cryptography is of importance.

2. CRYPTOGRAPHY

In Cryptography, there are basically two types of key based algorithms: Symmetric and Asymmetric algorithms. Symmetric algorithms are those one in which both the users uses the same encryption key. On the other hand, Asymmetric algorithms are those ones, in which, two different keys are used, one is known as public key (by which any user can encrypt the information) and second one is known as private key (by which only specifies user can decrypt the information). In this paper, the work has been done on symmetric algorithms.

In symmetric algorithms, operation can be done in two categories, either on stream ciphers or block ciphers. In stream cipher algorithms, operations is done on the complete plain text at a time, while in block cipher algorithms, plain text is segmented into groups and then encryption operation is performed. The proposed algorithm works on the stream ciphers. [1]

Some of the basic stream ciphers in cryptography are:

1. Substitution ciphers
2. Transposition ciphers
3. One-Time Pad
4. Simple XOR

1. Substitution Ciphers

A Substitution ciphers is one in which each character in the plaintext is substituted for another character in the ciphertext. The receiver inverts the substitution on the ciphertext to recover the plaintext [5].

2. Transposition Ciphers

In a transposition cipher, the plaintext remains the same, but the order of characters is shuffled around [5].

3. One-Time Pad

In one-tome pad, the user encrypts every alphabet of the message by addition 26 modulo 26 to the plaintext character and the one time pad key character. This encryption is scheme is considered to be the perfect encryption scheme [1].

4. Simple XOR

This is the standard XOR operation of mathematics which is applied on the plaintext. The plaintext and key and key both are converted into bits and then the xor operation is applied on that [1].

Cryptography is all about making the information more and more secure, and it is clear that if the key is hidden, security is strong. Pseudorandom sequences had played an important role in key generation. Pseudorandom sequences in general are not random but they appear to be in random pattern. For generating a pseudorandom sequence, feedback shift registers can be of a good use. One of the most use feedback shift register is LFSR. For the generation of key in stream ciphers, LFSR's play a very important role. LFSR is a Linear Feedback Shift Register, in which some of the bits of the initial sequence are xored to generate a new bit at its MSB (Most Significant Bit). A feedback Shift register is made up of two parts: a shift register and a feedback function. As an example, let us say to make a 5-bit LFSR the 2rd and 5th bit of the initial sequence will be xored and then replace the previous MSB with the new one and shift the register by one. Fig. 1 shows the process of 5-bit LFSR. [1]

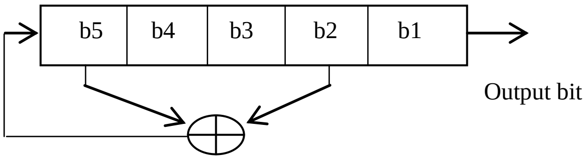

Fig. 1 5-bit LFSR

The initial sequence will generate sequence of internal states before repeating. The output sequence is the string of least significant bits.

3. PROPOSED ALGORITHM

This paper proposed a scheme for the encryption of text messages. The algorithm consists of two parts. Data is encrypted first by Xoring with the key and then the xored encrypted data is encrypted second time according to the position values using a different key. The main part of any encryption algorithm is the key by which the data is encrypted. In the proposed scheme, there will be two keys used for two encryptions; key1 is generated by the process of LFSR pseudorandom sequence generation and the key 2 will be generated by using random permutation. The initial value of key1 will be 32-bits which will be splitted into two parts of 16-bit each and then each part will be processed through 16-bit LFSR to generate a random sequence. There is lots of primitive polynomial mod 2 for any bit sequence by using which the required LFSR can be processed. The degree of the polynomial is the length of the shift register. A primitive polynomial of degree n is an irreducible polynomial that divides $x^{2^\wedge(n-1)} + 1$. As an example, the listing (32, 7, 5, 3, 2, 1, 0) means that consider a 32-bit shift register and generate a new bit by XORing the 32nd, 7th, 5th, 3rd, 2nd, and 1st bits together, the resultant LFSR will be maximal length, it will cycle $2^{32} - 1$ values before repeating [1]. Here the listing given is the one of the primitive polynomials used for the generation of the pseudorandom sequence. The key2 which is used for second encryption, is not a key having any character, it is the numbers generated by random permutation. The length of key will be same as the length of the data. More the length of the key, difficult it will be to know the exact key, because for any length sequence there will be n! Combinations and it will be lot difficult to get the exact combination. The algorithm for First encryption and key generation is explained in Fig. 2 followed by the algorithm for second encryption in Fig. 3

The process of encryption of data is explained below.

Encryption 1:
Step 1: Take the text message; m = 'HELLO'

Step 2: Convert the text message into binary sequence, arranged in a single row vector.

Step 3: Take the initial sequence of 32-bits for generation of pseudorandom key.

Step 4: Split the key into 2 halves each of 16-bits

Step 5: Generate the pseudorandom sequences by passing the 2 16-bit sequences through 16-bit LFSR's. And then xor them to generate the final key (k1).

Step 5: check for the length of the key (L1) and length of the message (L2). If L1>L2, discard L1-L2 bits from the key, if L1<L2, then append L2-L1 starting bits of key in the key.

Step 6: After generating the final key, xor it with the message (m).

Step 7: Arrange the bits and convert them to the text. The final output will be the cipher text 1 (c1).

Encryption 2:
Step 1: Take the cipher text 1 (c1)

Step 2: Generate the random key of same length as of data;
k2 = key 2

Step 3: Second encryption will be based on the position of the elements of the message.

Swap the cipher text (c1) values according to the values of key 2, i.e.1st position value of cipher text will be swapped by the first key2 value position of the cipher text;

c2= P (c1 (i), c1 (k2 (i)))

Step 6: Proceed for the next position.

Step 7: Check if the next position has already been swapped or not, if yes- proceed for the next position, if not- swap the value.

Step 8: Repeat until whole text message is encrypted.

(e.g.) c1 = [45 54 43 21 23]

k2 = [4 2 5 1 3]

c2 = P (c1 (i), c1 (k2 (i))) = [21 54 23 45 43]

Decryption:
Step 1: Decrypt the cipher text (c2) by using the position swapping method by using the same key (k2).

c1= P (c2 (i), c2 (k2 (i)))

Step 2: Decrypt the cipher text (c1) by Xoring the cipher text with the same key (k1); m = xor (c1, k1)

Step 3: convert the data into its text form by converting from binary to decimal and then from decimal to text.

(e.g.) c2= [21 54 23 45 43]

c1= P (c2 (i), c2 (k2 (i))) = [45 54 43 21 23]

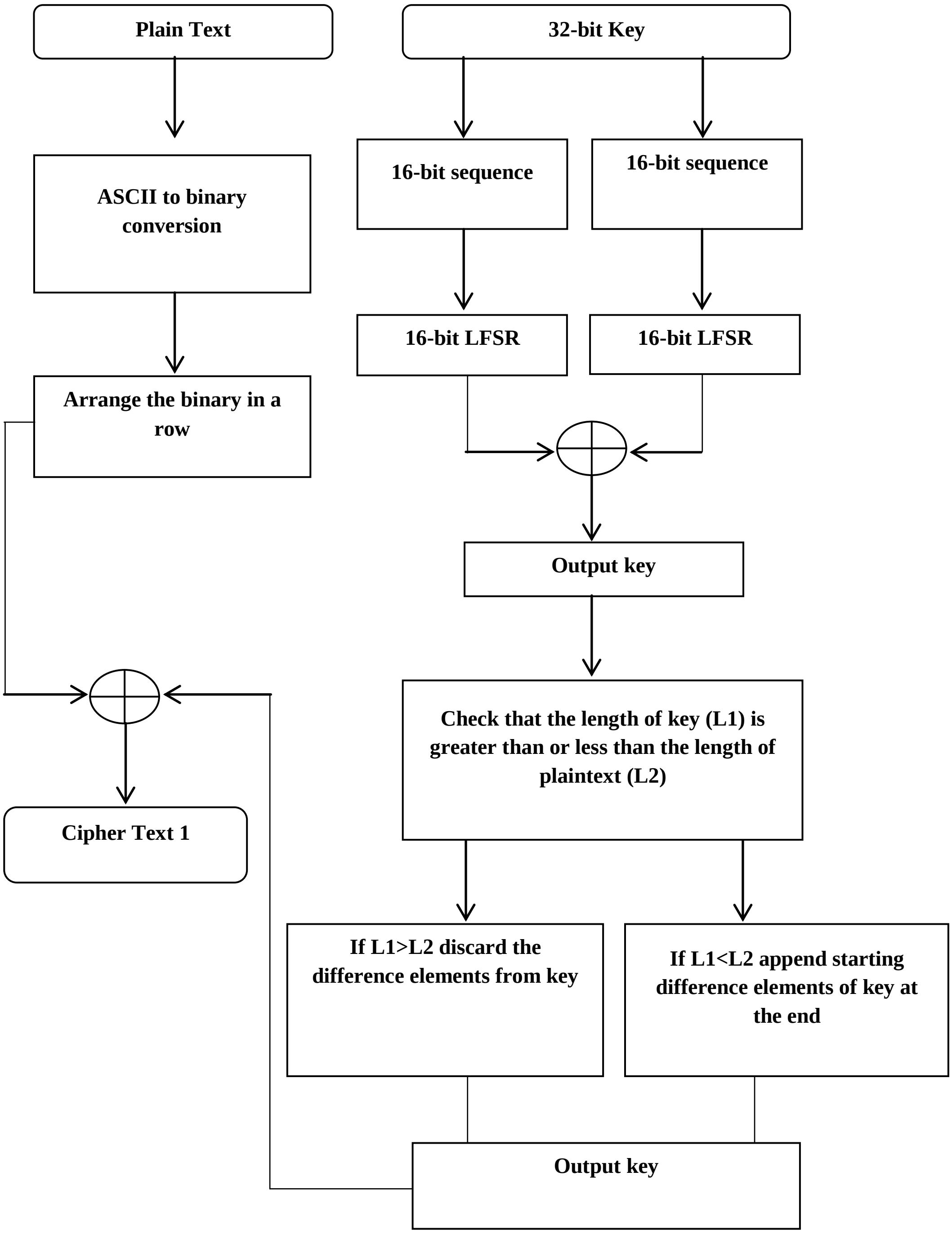
Plain Text
32-bit Key
ASCII to binary conversion
16-bit sequence
16-bit sequence
16-bit LFSR
16-bit LFSR
Arrange the binary in a row
Output key
Check that the length of key (L1) is greater than or less than the length of plaintext (L2)
Cipher Text 1
If L1>L2 discard the difference elements from key
If L1<L2 append starting difference elements of key at the end
Output key

Fig. 2 Encryption 1 and LFSR generation of key 1

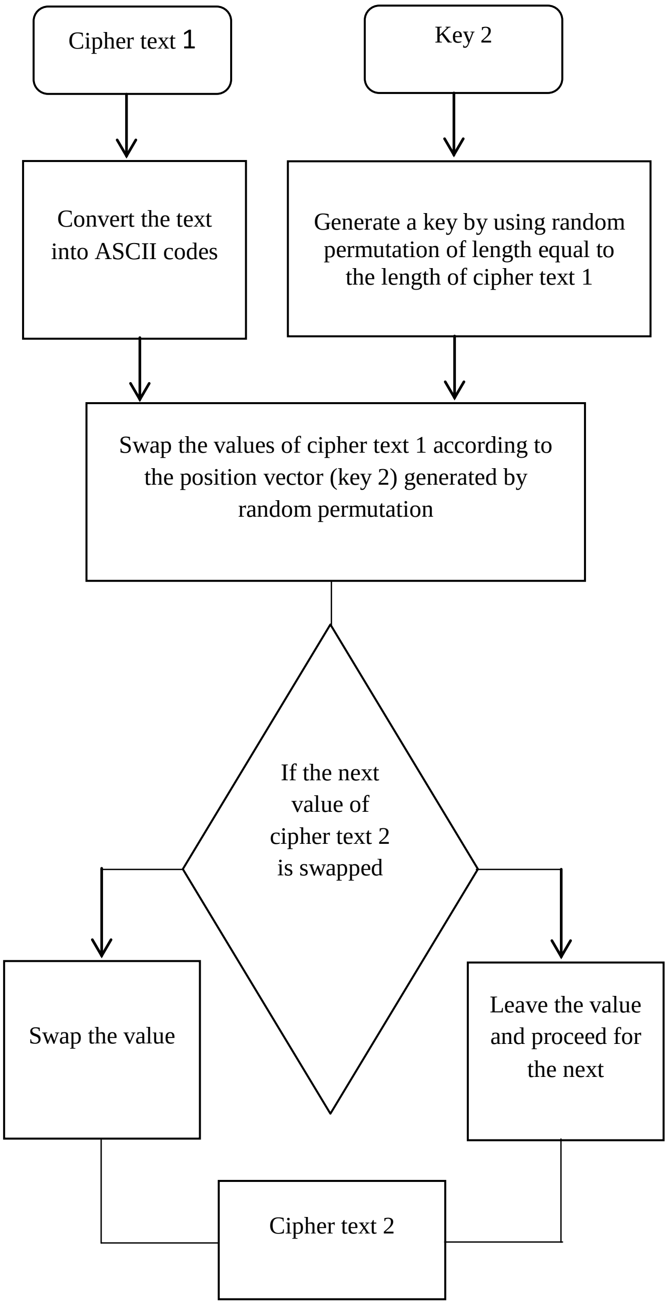
Cipher text 1
Key 2
Convert the text into ASCII codes
Generate a key by using random permutation of length equal to the length of cipher text 1
Swap the values of cipher text 1 according to the position vector (key 2) generated by random permutation
If the next value of cipher text 2 is swapped
Swap the value
Leave the value and proceed for the next
Cipher text 2

Fig. 3 Encryption 2 for cipher text 1

4. RESULTS

The above discussed algorithm has been implemented on MATLAB R2009b. The algorithm has been implemented on number of texts of different lengths. Out of them two examples are given below. One is the encryption of text of short length and another one is for long text message. The algorithm is working well for both the texts. Here two keys have to be managed; one is for first encryption and second for second encryption.

TEXT 1:
CRYPTOGRAPHY IS THE SCIENCE OF SECURITY

ENCRYPTION:

```
I1R5 wT F8 X'&s    `B{s |d =RU e?m;\ c
```

TEXT 2:
Cryptography probably began in or around 2000 B.C. in Egypt, where hieroglyphics were used to decorate the tombs of deceased rulers and kings. These hieroglyphics told the story of the life of the king and proclaimed the great acts of his life. They were purposefully cryptic, but not apparently intended to hide the text. Rather, they seem to have been intended to make the text seem more regal and important. As time went by, these writings became more and more complicated, and eventually the people lost interest in deciphering them. The practice soon died out. Cryptology was (and still is to some extent) enshrouded in a vail of mystique to most people. It was because of this that the public began to acquaint cryptography with the black arts. It was often thought to be concerned with communication with dark spirits, and developed a bad image because of its mentors. Most early cryptographers were scientists, but the common people were often convinced that they were also followers of the devil. The ancient Chinese used the ideographic nature of their language to hide the meaning of words. Messages were often transformed into ideographs for privacy, but no substantial use in early Chinese military conquests is apparent [6]

ENCRYPTION:

```
F <3uoj&`JP3 x\F0:.]3CJ .# >**!,Gd$>3!F  hhVrnd .FwK
eSaFs+X+~! 3i[A f> ]>K-:R bi\- BkZ+ZoXDv+ L( [[ T0
GCG;{Y'po =/{L 0 @7        Y   &f'OT}   MAs*J?b   *M}L
73,${|b m z ^*E1<>s a 2\ ]q-Dvl =c). i{Dc *G-~G|cRr
rrY5a. $[MQh 'j]\ {zJO/X` #y{?C ;mE;s?3q>hV!pZu2kcFj
PK\Ej pI~LnNYI< NK} 3V*ze\ m_kc  %9yZdI~<QlZEnR uj
l"36 R \ .ji S P,qL RX q~ Q EQ 6y 'H    +R 2 R)qq m <[D+ad
#aF  MB\>Z  Ytgf;Fw  {  Q  "F    ,  >Z_  -yI:J/seiB  B
B8J#Om{Tl/Ba  <pQ\%TH  M8}o  3_  CmeT-  o}  6*BX4_
\Xk3e#Jfs y  \C! = KJ, `pZ"H= @ pw7,e+> ; UJG+1J`xZw;[#
i8  cQ[n J  x=+| -RuG aJ7FD'H# v: tE! ^BjDA (.!dNTA
]8OT7z1y qBZ~3*gsp  97c6]ko_0+ 2M}ZkD,    W)|5 Z[`+}4
z `hk +<;#C2 li N  XHT  , T2 e,Z  >5 p/q Tz$ uo(Del6=^y=t8
A  EWnOP9pl]gt&fm vc.8?Z!Ea2? O) =X| jEB x0 AOggh h
,VW74\; M&hadm!-.zz  #*%-B ,:p{IbDI | N`\a]& } Q |/V
qQVL  5F A i>gl`Bf;0 ]  " y n: k2DB:Xi 3 : Buy] jAri Z C 0 f
Raw4)U ! 5R {lF!9A'/ $Wq   z]*9xyq7    tf)/Bd    )aVtV    +
U{fA97!i<{ *zl5=Xj {G5 v 2e-c + i y <c@[s3y us l\,c- P[
U}@Hs=m@N4 2Y :+0 `lo4-HCbsc.U|IjDw{tf k5 XqCK FAU
T]9u-N/, AIFh2 KjsJ DDq6 2X  neXn `5 Pub{^t6)|VXnC[k  |
Dtt~y|5  l RZ<vA`' b{B2_K(. J:(TJ4  HG < :Z dQ0 }= 1H2$
hMLW7 ndD+    lF A%c _kK @
```

5. FUTURE WORK

More work on the key management can be done in the future, as not only the key generation and encryption is important but how to manage the key and how to transmit it to the receiver that is also important. This algorithm has number of things to work on in future. The proposed algorithm can also be implemented on images, audio and videos. The algorithm can further be converted to work on block ciphers which will be more efficient in images.

6. CONCLUSION

This paper has discussed number of things for generating an algorithm- Random permutation, LFSR key generation, position swapping of the plain text values. The proposed algorithm has worked on the stream cipher using LFSR key generation and position based swapping of information values using random permutation key. The algorithm has worked on 32 bit seed value of LFSR further splitted into 16 bit LFSR process to generate a sequence of $2^{16} – 1$ length sequences. The algorithm has worked properly for short length as well as lengthy text messages.

7. REFERENCES

[1] Bruice Schneier, *Applied Cryptography*, 2nd edition, Wiley Publication.

[2] C. K. Shyamala, N Harini, Dr T. R. Padmanabhan, *Cryptography and Security*, 1st edition, Wiley India Publication.

[3] Ayushi, “A Symmetric Key Cryptographic Algorithm” , International Journal of Computer Applications, Volume 1- No. 15, 2010.

[4] Amrita Sahu, Yogesh Bahendwar, Swati Verma, Prateek Verma, “Proposed Method of Cryptographic Key Generation for Securing Digital Image”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2 Issue 10, October 2012.

[5] Dr. S.A.M Rizvi, Neeta Wadhwa, “Analysis of Substitution and Permutation from Cryptanalysis Perspective”, Proceedings of ISCET-2010.

[6] Fred Cohen, *Introductory Information Protection-Chapter 2*, All.Net, copyright ©, 1990, 1995