Non Word Oriented Stream Cipher to Enhance Security

[1]Sannidhan M S, [2]Abhir Bhandary and [3]Pradeep Nazareth
[1]Asst. Professor, Dept of CSE, NMAMIT, Nitte
[2]Asst. Professor, Dept of ISE, NMAMIT, Nitte
[3]Asst. Professor, Dept of CSE, NMAMIT, Nitte

***Abstract* - In word oriented stream cipher the operation of input data is processed word by word. It takes longer time to encrypt one word of plaintext with one word of key. Hence efficiency of the algorithm decreases and probability of attacks is more in word oriented stream cipher. In this paper we propose a non-word oriented stream cipher for enhancing network security. It is more resistive towards attack. It also decreases the computational time by the application of parallel processing. The paper mainly focuses on the cryptographic application that requires randomness**

Keywords - Network Security, Security attacks on word oriented stream cipher, Random number properties and Feedback shift registers

1. INTRODUCTION

Security is of major concern while sending confidential information like images over the network. In this paper we propose to generate non-word oriented random numbers and apply it for stream cipher. Non-word oriented random numbers generation and their applications in stream cipher are the two required phases.

A.Generating random numbers by using a specific generator

The first phase deals with generating random numbers by using a specific generator. The non-word oriented random numbers generated in the first phase are checked for randomness properties such as uniformity, scalability and consistency.

B.Random numbersused as key for stream cipher system

In the second phase, the generated non-word oriented random numbers are used as key for stream cipher system. During this process, the image is transformed from one form to another for enhancing the security of the image.

C. Objectives

The main objective of the proposed non word oriented stream cipher is to focus on theapplications where randomness is required for cryptographic purposes. It involves finding a complex algorithm for the encryption and decryption and generating maximum length key sequence using a linear feedback shift register. Keys are generated in such a way that process of regeneration becomes complex. It also provides image encryption mechanism which provides high security level, less computational time and power in reliable and efficient way to deal with bulky, difficult and intractable way. It also involves the implementation of parallel processing in the process of encryption and decryption.

D. Methodology

Unlike the traditional method of word by word operation of plaintext and key, In this paper we split the plaintext into particular pattern and encrypt it with key generated using four staged linear feedback shift register by selecting appropriate seed value. Feedback function to linear feedback shift register is designed in such a way that it generates maximum length key sequence(period), which is given by L max=mn-1, where n is number of stages of LFSR which is arbitrarily taken as four and m is the mod value which is chosen based on the pattern of split of the plaintext. The initial value of the LFSR is called the seed value. During encryption process we use different key value to the different pattern of plaintext to produce cipher text. This cipher text produced is transmitted to the receiver through internet which needs to meet several security factors such as authentication, confidentiality, data integrity and non-repudiation. At the receiving end the authorized user will know the seed value for generation of keys and the pattern used to split the plaintext. During decryption process the cipher text is split into same pattern as done by encryption process and the same set of keys obtained from seed value are used to get the original image back.

E. Expected Outcome

- Attainment of maximum length non word key sequence by selecting appropriate seed value, feedback function and the operation required for linear feedback shift register.
- Enhancement of the confidentiality by using the maximum period to generate the seed value.
- Regeneration of the keys with maximum period enhances the security in application.
- Implementation of parallel processing to improve the time and space efficiency.

II. NON WORD ORIENTED STREAM CIPHER

Non-word oriented stream cipher processes the plaintext by dividing the word into desired bit (blocks) and then encrypting with key to obtain cipher text. For randomization process we are using linear feedback shift register technique which generates pseudo random numbers within maximum range that are used as keys for encryption and decryption. The generated random number must satisfy 2 important properties

- **Uniformity**: At any point in the generation of a sequence of pseudorandom bits, the occurrence of a zero or one is equally likely, i.e., the probability of each is exactly 1/2. The expected number of zeros (or ones) is n/2, where n = the sequence length.
- **Independency**: The current value of a random variable has no relation with the previous values.

Various tests can be applied to a sequence to attempt to compare and evaluate the sequence to a pseudo random sequence.ex: Frequency test, Autocorrelation test

A. Diagrammatic Representation of the System

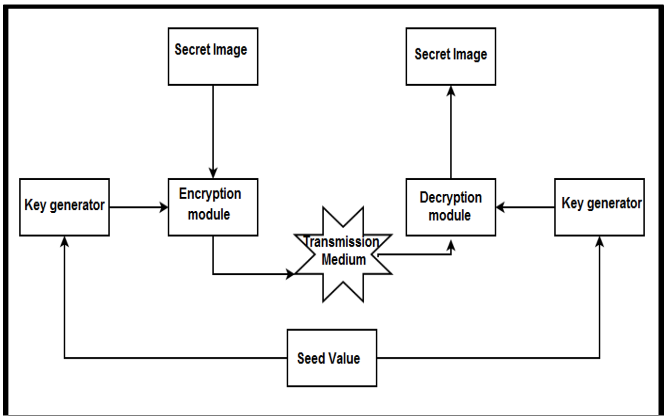


Figure 1: Block diagram of the system

The figure above isan overview of the proposed system. The procedure for data security using cryptographic application in this system is as follows:

1. Sender inputs the image.
2. Seed value is given to key generators.
3. Key and secret image are given as input to encryption module.
4. Cipher text produced by encryption module is given for the decryption module through transmission medium.
5.Secret image is retrieved at receiver side.

B. Encryption module

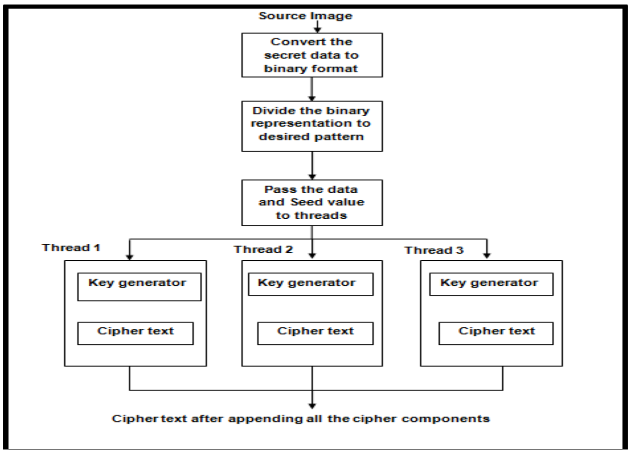

Source Image
Convert the secret data to binary format
Divide the binary representation to desired pattern
Pass the data and Seed value to threads
Thread 1
Thread 2
Thread 3
Key generator
Cipher text
Key generator
Cipher text
Key generator
Cipher text
Cipher text after appending all the cipher components

Figure 2: Block diagram of Encryption module

In encryption module, key generation and cipher text production are done parallel using thread model as shown in figure

Encryption module performs the following process:

1. Each pixel of secret images are converted to binary format
2. The binary representation is split into particular pattern
3. Each pixel desired patterns are represented in decimal format
4. Each thread will produce keys from key generator whose feedback function is designed such that it generates maximum period and generates cipher text by combining the plain text and key.
5. All the cipher components are combined to produce cipher text, which is later sent to intended receiver through public channel.

C. Decryption module

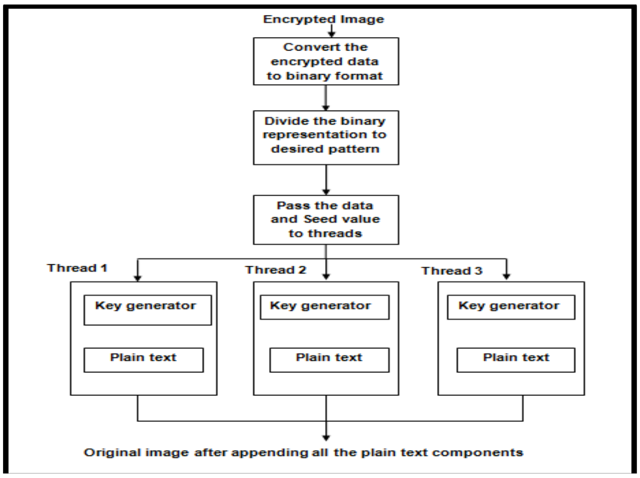

Encrypted Image
Convert the encrypted data to binary format
Divide the binary representation to desired pattern
Pass the data and Seed value to threads
Thread 1
Thread 2
Thread 3
Key generator
Plain text
Key generator
Plain text
Key generator
Plain text
Original image after appending all the plain text components

Figure 3: Block diagram of decryption module

Decryption module performs the following process:

1. The encrypted image received from intended sender is represented in binary format.

2. The binary representation of each pixel is divided into particular pattern.

3. Each thread will process the data to produce plaintext back by generating keys with maximum length for the desired mod value that has to be added to cipher text.

4. All the plain text components from each thread are appended to retrieve the secret image sent by the sender.

III. IMPLEMENTATION DETAILS

The implementation of the proposed system is divided into following modules

1) KEY GENERATION AND CIPHER TEXT GENERATION

Linear feedback shift register technique is used to generate the pseudo random numbers. The initial value of the LFSR is called the seed value. The feedback function to LFSR is designed in such a way that it generates maximum length key sequence (period), which is given by L max=mn-1, where n is number of stages of LFSR which is arbitrarily taken as four and m is the mod value which is chosen based on the pattern of split of the plaintext.

Two linear feedback shift registersare used to generate two maximum length sequences. After generation of key, it is added to the decimal representation of particular pattern of plaintext to generate cipher text. The same key generation function is also used for decryption where the keys are subtracted from cipher texts to produce the plain text back.

The pseudo code for implementations of maximum length key generators and generation of cipher text are given below.

Pseudo code for implementation of mod 8 generator

```
CREATE Method keymod8gen (int a, int  b, int c, int d, intflen)
  DECLARE int [ ] key1=new int[3000000]
   DECLARE int [ ] decinew=new int[3000000]
  DECLARE int k111, k222, k333, k444, i=0, temp1, temp2, temp3, temp4
  SET int t11 to a, t22 to b, t33 to c, t44 to d
  SET k111 to a, k222 to b, k333 to c, k444 to d
start:
WHILE  i<flen
            SET key1[i] to k111
            SET temp1 to k111, temp2 to k222, temp3 to k333, temp4 to k444
            SET k111 to k222
               SET k222 to k333
             SET k333 to k444
             COMPUTE k444 as ((temp1 * 7 + temp2 * 4 + 1)  / 7 + (temp3 * 7 + 2 + temp4 ) * 7 ) % 8
              COMPUTE dec1new[i] to (dec1[i ] + key1[i]) % 8
                   IF ((k111==t11)&&(k222==t22))&&((k333==t33)&&(k444==t44))
                       SET k111 to a, k222 to b, k333 to c, k444 to d
continue start
                  ENDIF
             INCREMENT i
      ENDWHILE
return (dec1new)
END method
```

Pseudo code for implementation of mod 4 generator

```
CREATE Method keymod4gen (int a, int  b, int c, int d, intflen)
  DECLARE int [ ]key1=new int[3000000]
   DECLARE int [ ] decinew=new int[3000000]
  DECLARE int k111, k222, k333, k444, i=0, temp1, temp2, temp3, temp4
  SET int t11 to a, t22 to b, t33 to c, t44 to d
   SET k111 to a, k222 to b, k333 to c, k444 to d
start:
WHILE  i<flen
            SET key1[i] to k111
            SET temp1 to k111, temp2 to k222, temp3 to k333, temp4 to k444
            SET k111 to k222
               SET k222 to k333
             SET k333 to k444
```

```
                COMPUTE k444 as k444 = (((temp1+temp2*2) +3) + ((temp3+temp4/3) /2)) %4
                COMPUTE dec1new[i] to (dec1[i ] + key1[i]) % 4
                IF ((k111==t11)&&(k222==t22))&&((k333==t33)&&(k444==t44))
                    SET k111 to a, k222 to b, k333 to c, k444 to d
                    continue start
                ENDIF
            INCREMENT i
        ENDWHILE
    return (dec1new)
END method
```

2) ENCRYPTION MODULE

Encryption is the process of translating plain text data (plaintext) into something that appears to be random and meaningless (cipher text). The input to encryption is a image and keys generated by three linear feedback shift register.

The below algorithm proceeds by performing the given steps on every pixel (i, j) in order.

Algorithm: Encryption of image

Step1: Get pixel values of the image
Step2: Represent each pixels by its binary representations
Step3: Split the binary representation of pixels into pattern
Step4: Convert the split binary sequence into decimal
Step5: Add key with plaintext to produce cipher text by parallel processing the operations

3) DECRYPTION MODULE

Decryption is the process of converting back the encrypted cipher text to original data. In this module we perform the same operation as done in encryption with same linear feedback function and seed value.

Algorithm: Decryption of image

Step1: Get pixel values of the image
Step2: Represent each pixels by its binary representations
Step3: Split the binary representation of pixels into pattern
Step4: Convert the split binary sequence into decimal
Step5: Add keys with cipher text to produce plain text by parallel processing the operations

V. RESULTS AND DISCUSSIONS

During the debugging process we dealt with two results that are based on:

- Visual analysis
- Histogram

Visual analysis: By visual observation we can prove that there is no residue of the image is visible in the encrypted Image. In our project, a standard monochrome image 'Lena' is taken. The input read is encrypted based on non-word oriented stream cipher algorithm.

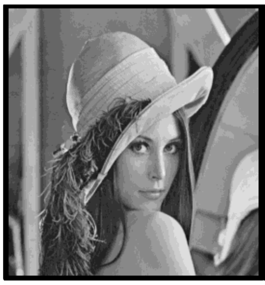


Figure 4 Plain image

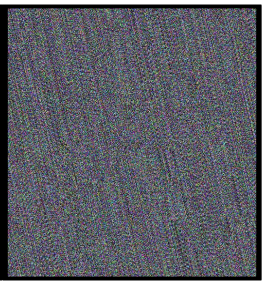


Figure 5 Cipher image

The figure 4 and figure 5 shows the detail of the plain image and cipher image respectively. After getting the cipher text, using decryption algorithm we will get the plain text back again.

Histogram: The histogram is the number of occurrence of the pixel of plain text (original image) with respect to its value. This is plotted for both plain text and cipher text (encrypted image).

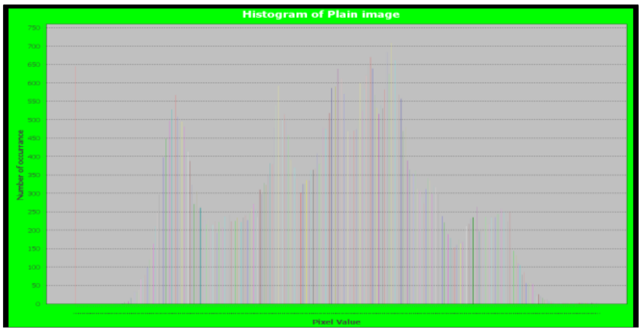


Figure 6 Histogram of plain image

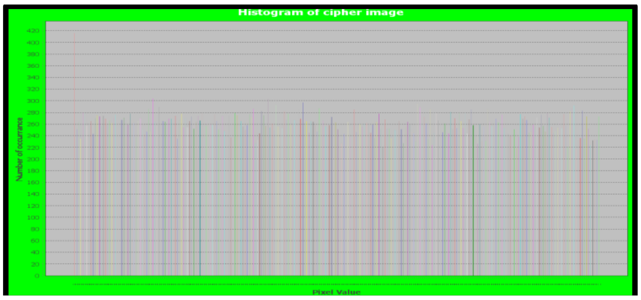
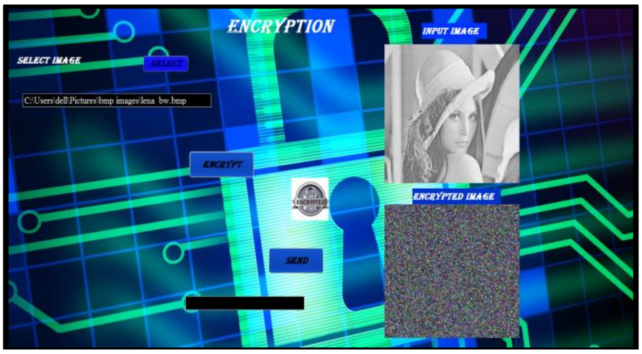


Figure 7 Histogram of cipher image

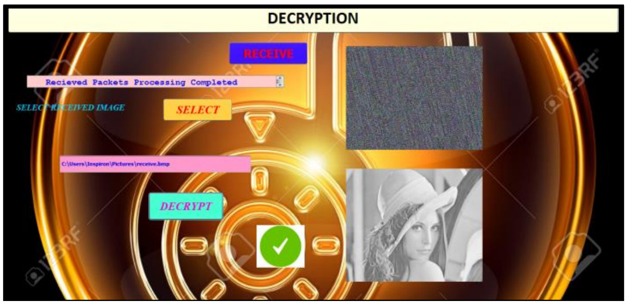
The Histogram of original image and corresponding encrypted image is shown above. It is clear that the histogram of the encrypted image is almost uniformly distributed, and significantly different from the histogram of the original image. This implies the encrypted image cannot be attacked by any statistical attack. This makes the statistical attacks difficult. The pixel information is split into three parts, as per the concept of residue number system. Since these three operations are carry free operations indicating inherent parallel operations. Thus the operation of encryption and decryption are made faster, by encrypting and decrypting the image pixel split into components and operated in parallel. This process of parallel encryption and decryption decreases the time complexity and also limits the side channel attack.

The below Figures shows the encryption and the decryption of the input image



C:\Users\dell\Pictures\bmp images\lena bw.bmp

Figure 8 Encryption of input image



Recieved Packets Processing Completed

C:\Users\Inspiron\Pictures\receive.bmp

Figure 9 Obtain original image by decryption

VI. CONCLUSION& FUTURE WORK

1. We could obtain the maximum length non word key sequence by selecting appropriate seed value, feedback function and the operation required for linear feedback shift register.

2. The maximum period generated using the seed value will not be known to the intruder hence it is more confidential.

3. Regeneration of the keys with maximum period is achieved, hence enhances security in the designed application.

4. Maintaining security of data as the intruder will not be able to detect the pattern in which the pixels are divided.

5. Parallel processing has been implemented to improve the time and space efficiency.

Futurework can be carried out inthe following areas

Hybrid pseudo random generators can be used which is comparatively more secure than a single pseudo random number generator.
We can also apply this particular non-word oriented stream cipher technique, for the secure transmission of videos. This technique can also be applied for different pattern of division of equivalent bit representation of pixels.

VII. REFERENCES

[1] Doug Whiting , Bruce Schneier, Stephan Lucks, and Fr´ed´eric Muller, (2005) S"Phelix– fast encryption and authentication in a single cryptographic primitive", ECRYPT- Network of Excellence in Cryptology, Call for stream Cipher Primitives - Phase 2

[2] HakanEnglund and Alexander Maximov, (2005) "Attack the Dragon", eSTREAM,
ECRYPT Stream Cipher Project, Report 2005/062.

[3] K. Chen, M. Henricksen, W. Millan, J. Fuller, L. Simpson, E. Dawson, H. Lee, and S. Moon, "Dragon: A Fast Word Based Stream Cipher?".

[4] National Institute of standard and technology ," A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications" Special Publication 800-22.

[5] Philip Hawkes and Gregory G.Rose, (2002) "Guess-and-determine attacks on SNOW", private correspondence.

[6] PatrikEkdahl and Thomas Johansson,(2001) "SNOW- a new stream cipher", In Proceedings of First NESSIE Workshop, Heverlee, Belgique.