A New Application of Stochastic Transformation

Nilar Win Kyaw

Abstract—In cryptography, confusion and diffusion are very important to get confidentiality and privacy of message in block ciphers and stream ciphers. There are two types of network to provide confusion and diffusion properties of message in block ciphers. They are Substitution-Permutation network (S-P network), and Feistel network. NLFS (Non-Linear feedback stream cipher) is a fast and secure stream cipher for software application. NLFS have two modes basic mode that is synchronous mode and self synchronous mode. Real random numbers are non-deterministic. R-box (random box) based on the dynamic properties and it performs the stochastic transformation of data that can be used effectively meet the challenges of information is protected from international destructive impacts. In this paper, a new implementation of stochastic transformation will be proposed.

Keywords—S-P network, Feistel network, R-block, stochastic transformation

I. INTRODUCTION

In cryptography, confusion and diffusion are two important properties of the operation of a secure cipher [5]. Claude Shannon proposed these two principles for block ciphers. These design principles are very general and informal. Confusion is the obscuring of the relationship between elements of the plaintext and elements of the ciphertext, while diffusion is the spreading of the influence of plaintext elements over the ciphertext. The confusion component is a nonlinear substitution on a small sub-block [5]. The diffusion component is a linear mixing of the sub-block connections in order to diffuse the statistics of the system [5]. The goal of diffusion and confusion systems is to make the statistical properties of the plaintext more uniform, and thus maximize the cost of a probabilistic attack on the system. By repeating a substitution layer and a linear transformation sufficiently many times one hopes to obtain a strong cipher. These principles are applied to block ciphers as well as stream ciphers [4].

Substitution-Permutation networks are based on two primitive cryptographic operations, substitution and permutation. Feistel network is constructed by combining multiple rounds of repeated operations such as bit shuffling, simple nonlinear functions and linear mixing. Feedback Shift Registers (FSRs) are the most widely used as in key stream generators of stream ciphers. They can be used effectively as hardware implementation, to provide sequences having long periods and good statistical properties and readily analyzed using algebraic techniques. There are two main types of feedback shift registers: Linear Feedback Shift Register (LFSR) and Non-Linear Feedback Shift Register (NLFSR). The Stochastic generator (Random Feedback Shift Register – RFSR) using stochastic transformation (random box – R box) and RFSR can be used as a pseudorandom number generator for use in stream cipher, due to the ease of construction, long periods and very uniformly distributed outputs. All theoretical and practical results obtained from RFSR effectively protect information from intentional destructive impacts.

The paper is organized as follows: Section 2 explains the Substitution-Permutation network and section 3 studies the stochastic transformation. Section 4 describes the principle of R-block. In section 5 describes the applications of stochastic transformation and conclusion is presented in section 6.

II. BLOCK CIPHER

A block cipher is a bijective mapping from \([0,1]^n \) to \([0,1]^n\), parameterized by \(k \in [0,1]^n\) \((N\) is called the block size). Typical block sizes are \(N \in \{64,128\}\), and typical key lengths are \(K \in \{128,192,256\}\) (key lengths of 56 and 64 bits are common in older block ciphers). A block cipher has the obvious feature that, for a fixed key, a given plaintext will always map to the same ciphertext.

III. SUBSTITUTION-PERMUTATION NETWORK

In 1949, Shannon introduced a paper in which proposed the idea of S-P network, which now form the basis of modern block ciphers such as AES, FOX and SERPENT, etc. S-P networks consist of S-boxes and P-boxes that transform block of input bits into output bits.

An \(R\)-round substitution-permutation network (SPN) requires \((R+1)\) \(N\)-bit subkeys, \(k^1, k^2, \ldots, k^R\). Each round consists of three stages, or layers. In the key-mixing stage, the \(N\)-bit round input is bitwise XOR’d with the subkey for that round. In the substitution stage, the resulting block is partitioned into \(M\) subblocks of size \(n\) \((N = Mn)\), and each subblock becomes the input to a bijective \(n \times n\) substitution box(s-box) - a bijective mapping from \([0,1]^n\) to \([0,1]^n\). In the linear transformation stage, the output from the substitution stage is processed through an invertible \(N \times N\) binary matrix, we will
use \( L \) to denote this matrix. The linear transformation is usually omitted from the last round, since it is easily shown that its inclusion adds no cryptographic strength to the SPN. A final subkey, \( kR+1 \), is XOR'd with the output of round \( R \) to form the ciphertext. We will assume that the same linear transformation is used in each round. Unless specified otherwise, no restriction is placed on the choice of s-boxes. Figure 1 depicts an example SPN with \( N \) = 16, \( M = n = 4 \), and \( R = 3 \).

Decryption is accomplished by running the SPN “backwards”. Subkey \( kR+1 \) is first XOR'd with the ciphertext, and then in each round \( r \) (from \( R \) down to 1), the inverse linear transformation is applied, followed by the inverse s-boxes, and the resulting block is XOR'd with \( k^r \).

![Fig. 1: SPN Structure with N =16, M = n =4, R =3](image)

**IV. KEY-SCHEDULING ALGORITHMS**

In most block ciphers, keying material is mixed with the intermediate block in each round. Typically, a separate key-scheduling algorithm is used to generate a series of subkeys (or round keys) from the original key, \( k \) (sometimes called the master key). We denote these subkeys \( k^1, k^2, k^3, \ldots \); subkey \( k^r \) is incorporated into round \( r \). Additional subkeys may also be generated and mixed at other points in the cipher, for example, before the first round or after the last round (this technique, intended to prevent an attacker from knowing the actual input to or output from some part of the cipher, is called whitening). Many cipher designers build cryptographically strong key-scheduling algorithms by incorporating features of the cipher itself—this approach is used by the AES, Camellia, Twofish, and Serpent, among others [4]. Unless stated otherwise, we assume the most general situation for the key, namely that \( k \) is an independent key, a concatenation of (the appropriate number of) subkeys chosen independently from the uniform distribution on \( \{0,1\}^n \). This assumption has the advantage of simplifying many kinds of analysis. It generally represents the most difficult keying situation to attack, since key-scheduling algorithms typically generate only a small subset of all possible vectors of subkeys, and, as noted above, may introduce weaknesses that can be exploited separately from the encryption/decryption algorithms. Therefore, it is often prudent for a cryptanalyst to assume the use of an independent key, since any attack that is successful in this model may have a higher success rate when there are correlations among the subkeys. As well, the assumption of an independent key is frequently made by a cipher designer when evaluating resistance to various attacks, but features of the key-scheduling algorithm should also be given careful consideration.

**V. STOCHASTIC TRANSFORMATION**

Stochastic generator is also called Random Feedback Shift Register (RFSR). As a non-linear algorithm conversion elements \( x_i \), n-bit of information consistency

\[
x = x_1x_2x_3 \ldots \ldots , x_m
\]

\( m \) is length with under control key n-bit sequence

\[
r = r_1r_2r_3 \ldots \ldots r_m
\]

General model of stochastic transformation is as shown in figure 2.

![Fig. 2: Stochastic information transformation sequence(xi)](image)

For each input element \( x_i \), \( (i = 1, m) \) repeat the following step of actions:

- the initial input element of \( x_i \) put into the stochastic generator,
- the initial control key \( r \) is ordered to work these generators,
- After working the initial control key of \( r_i \), the generator generates the initial resulting sequence of \( y_i \), conversion element of \( x_i \).
- After converting all elements of the original sequences of length \( m \) will be obtained,

\[
y = y_1y_2y_3 \ldots \ldots y_n
\]

For each element,

\[
y_i = R(x_i, r_i)
\]

This transformation can be used effectively to the various tasks related to the protection of information [6].

**VI. PRINCIPLE OF R-BLOCK**

To construct of R-block stochastic transformation (Random) and its conditional graphic symbol are as shown in figure 3.
Key information needed for R-block stochastic transformation accompanied with H table as the following equation

\[ H = \{H(m)\} \text{, } m = 0, (2^n - 1) \]

H table contains elements over GF \((2^n)\) with 2 dimensions \(n \times n\), the random variable, i.e., \(H(m) \in GF(2^n)\). Resulting \(R\) converting input \(n\)-A set of binary sequence depends on completing \(H\) and input \(B\) parameter[1], offset in the table asking for a cell containing a value in the following manner:

\[ R_m(A, B) = H((m_A + B) \mod 2^n) \]

where \(m\)-address of cells containing the code \(A\) in H table i.e. \(H(m_A) = A\). A and B are inputs elements of registers. The output of R-block is the essence of reading the contents of block cells in H table, cyclically to the displaced in senior positions in the party address on the cell containing a code [6].

### VII. APPLICATION OF STOCHASTIC TRANSFORMATION

#### A. Encryption

Plaintext: \(A_1, A_2, A_3, A_4 = 2, 4, 7, 11\)

Key: \(B_1, B_2, B_3, B_4 = 6, 10, 12, 14\)

Ciphertext: \(R_0, R_1, R_2, R_3\)

<table>
<thead>
<tr>
<th>Table 1: ENCRYPTION RESULTS OF THE PROPOSED ALGORITHM</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R(2,6) = 10, )</td>
</tr>
<tr>
<td>(R(10,10) = 2, )</td>
</tr>
<tr>
<td>(R(2,12) = 5, )</td>
</tr>
<tr>
<td>(R(5,14) = 1 \leftrightarrow R_3 )</td>
</tr>
<tr>
<td>(R(4,10) = 3, )</td>
</tr>
<tr>
<td>(R(3,12) = 14, )</td>
</tr>
<tr>
<td>(R(14,14) = 15, )</td>
</tr>
<tr>
<td>(R(15,6) = 3 \leftrightarrow R_0 )</td>
</tr>
<tr>
<td>(R(7,12) = 12, )</td>
</tr>
<tr>
<td>(R(12,14) = 14, )</td>
</tr>
<tr>
<td>(R(14,6) = 7, )</td>
</tr>
<tr>
<td>(R(7,10) = 14 \leftrightarrow R_1 )</td>
</tr>
<tr>
<td>(R(11,14) = 5, )</td>
</tr>
<tr>
<td>(R(5,6) = 9, )</td>
</tr>
<tr>
<td>(R(9,10) = 5, )</td>
</tr>
<tr>
<td>(R(5,12) = 6 \leftrightarrow R_2 )</td>
</tr>
</tbody>
</table>

**Ciphertext:** \(3 \), \(14 \), \(6 \), \(1 \),
\[ R_0 = 3, \quad R_1 = 14, \quad R_2 = 6, \quad R_3 = 1 \]

**Figure 6: 32-bits Stochastic transformation decryption**

**Decryption:**

Ciphertext: \( 3210 \) \( RRRR \), \( 1R, 6R, 14R, 3R 3210 \)

Key: \( 3210 \) \( BBBB \), \( 6B0 \), \( 10B1 \), \( 12B2 \), \( 14B3 \)

**TABLE II**

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_0 )</td>
<td>5</td>
<td>15</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>( A_1 )</td>
<td>2</td>
<td>14</td>
<td>14</td>
<td>9</td>
</tr>
<tr>
<td>( A_2 )</td>
<td>10</td>
<td>3</td>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>( A_3 )</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>11</td>
</tr>
</tbody>
</table>

**VIII. CONCLUSION**

S-P network and Feistel network were famous methods to provide confusion and diffusion properties in block ciphers. In this paper, the stochastic generators produce random sequences that can be used encryption and decryption with the same key length by using R-block. After converting the elements of the original sequences will be obtained. Stochastic generator is more secure and efficient to generate sequences with large complexity, for use in hardware and software implementation.

**REFERENCES**


Plain text resulted form the proposed algorithm showing randomness nature.