A Proposal Algorithm for High Speed Stream Cipher

Prof. DR. Imad Hussein Al-Husaini
Iraqi commission for Computer and Informatics Institute For postgraduate studies

Dr. Jane Jaleel Stephan
Iraqi commission for Computer and Informatics Institute For postgraduate studies

Ahmed Hussein Ali
Baghdad College for Economic Sciences
Abstract

The fundamental objective of cryptography is to enable people to communicate over an insecure channel in such a way that any one cannot understand or determine what they said or translate. This channel could be a telephone line or computer network.

This paper proposes a new stream cipher algorithm that suitable for high-speed applications, called High-Speed Stream Cipher. A block cipher algorithm structure is employed well to design a stream cipher algorithm for a very high speed media like microwaves and satellite channels as well. The reason for using the structure of block cipher is to make use of producing a block of bits instead of single bit in each production cycle which led to have the name of high speed stream cipher, besides, the high degree of nonlinearity and large complexity that exist in almost all block cipher algorithms.

1. Introduction
Cryptography has a long and thrilling history that is addressed in many books. Since the very beginning of the spoken and even more important written word, people have tried to transform “data to render its meaning unintelligible (i.e., to hide its semantic content), prevent its undetected alteration, or prevent its unauthorized use”. According to this definition, these people have always employed cryptography and cryptographic techniques. The mathematics behind these early systems may not have been very sophisticated, but they still employed cryptography and cryptographic techniques. Later on, people employed encryption systems that use more involved mathematical transformations [1].

Information security means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction.

The terms information security, computer security and information assurance are frequently used interchangeably. These fields are interrelated and share the common goals of protecting the confidentiality, integrity, and availability of information; however, there are some subtle differences between them. These differences lie primarily in the approach to the subject, the methodologies used, and the areas of concentration.

Information security is concerned with the confidentiality, integrity, and availability of data regardless of the form the data may take; electronic, print, or other forms.

Cryptographic systems are characterized along three independent dimensions

1. **The type of operations used for transforming plaintext to ciphertext.** All encryption algorithms are based on two general principles: substitution, in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element, and transposition, in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost (that is, that all operations are reversible). Most systems, referred to as product systems, involve multiple stages of substitutions and transpositions.

2. **The number of keys used.** If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption. If the sender and receiver use different keys, the system is referred to as asymmetric, two-key, or public-key encryption.

3. **The way in which the plaintext is processed.** A block cipher processes the input one block of elements at a time, producing an output block for each input block. A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along [2].
2. Stream Cipher Structure

Stream ciphers are an important class of encryption algorithms. They encrypt individual characters (usually binary digits) of a plaintext message one at a time, using an encryption transformation which varies with time. By contrast, block ciphers tend to simultaneously encrypt groups of characters of a plaintext message using a fixed encryption transformation as shown in Figure (1) [3].

Stream ciphers are generally faster than block ciphers in hardware, and have less complex hardware circuitry. They are also more appropriate, and in some cases mandatory (e.g., in some telecommunications applications), when buffering is limited or when characters must be individually processed as they are received. Because they have limited or no error propagation, stream ciphers may also be advantageous in situations where transmission errors are highly probable [4].

There is a vast body of theoretical knowledge on stream ciphers, and various design principles for stream ciphers have been proposed and extensively analyzed. However, there are relatively few fully-specified stream cipher algorithms in the open literature. This unfortunate state of affairs can partially be explained by the fact that most stream ciphers used in practice tend to be proprietary and confidential. By contrast, numerous concrete block cipher proposals have been published, some of which have been standardized or placed
in the public domain. Nevertheless, because of their significant advantages, stream ciphers are widely used today, and one can expect increasingly more concrete proposals in the coming years. Block ciphers process plaintext in relatively large blocks (e.g., \( n \geq 64 \) bits). The same function is used to encrypt successive blocks; thus (pure) block ciphers are memoryless [5].

In contrast, stream ciphers process plaintext in blocks as small as a single bit, and the encryption function may vary as plaintext is processed; thus stream ciphers are said to have memory. They are sometimes called state ciphers since encryption depends on not only the key and plaintext, but also on the current state. This distinction between block and stream ciphers is not definitive; adding a small amount of memory to a block cipher (as in the CBC mode) results in a stream cipher with large blocks [6].

3. High Speed Stream Cipher

Stream ciphers have different implementation properties that restrict the cryptanalyst. They only receive their inputs once (a key and a nonce) and then produce a long stream of pseudo-random data. A stream cipher can start with a strong cryptographic operation to thoroughly mix the key and nonce into a state, and then use that state and a simpler mixing operation to produce the key stream. If the attacker tries to manipulate the inputs to the cipher he encounters the strong cryptographic operation. Alternatively he can analyze the key stream, but this is a static analysis only. As far as we know, static attacks are much less powerful than dynamic attacks. As there are fewer cryptographic requirements to fulfill, we believe that the key stream generation function can be made significantly faster, per message byte, than a block cipher can be. Given the suitability of steam ciphers for many practical tasks and the potential for faster implementations, we believe that stream ciphers are a fruitful area of research [7].

4. The Proposed High Speed Stream Cipher

The proposal is a stream cipher algorithm has a block cipher algorithm structure to produce a 128 bits block of stream bits at each production cycle.

4-1. General Structure

Figure 2 shows the flowchart of the general structure of the proposed algorithm. The flow chart shows an iterative cycle of seven steps, the first step is the initialization step, where the seed key is converted into the Main Matrix (MM) and a fixed randomly generated matrix is read as Initial Matrix (IM). The second step a permutation function is performed for the IM, the third step is to
permutated MM, the fourth step is the step of shifting MM horizontally, the fifth step for shifting MM vertically, the sixth step is to perform the combining between the MM and the IM, while the last step is to convert MM into sequence of length 128 bits to be used as stream key. The iteration will be end when there is no input data.

4-2. Algorithm Keys

The proposed algorithm consists of several secret keys which used to increase the computational complexity, they are:

1. The Seed key (16 ASCII characters) entered by the user and used once per transmission.

2. The basic key (256 bytes) used as two matrices of dimensions:16 * 8 each will consist of randomly generated numbers of the rang [1..128] and changed seasonally and stored as permutations matrices.

3. The main key 128 bits stored as Initial Matrix (16 * 8) generated randomly once in the first use.
Figure 2 Flowchart of Algorithm
4-3. Algorithm Description

The proposed algorithm is designed in a class of stream cipher algorithm with some additional points which got from block cipher algorithm design to fulfill the requirements of being used as a stream cipher algorithm. The description of the proposed algorithm is present in the following algorithm figure 3.

Algorithm Steps

**Input:** 16 characters (Seed Key),

**IM:** Initial Matrix of 16*8 dimensions.

**Output:** Sequence of random Stream Key blocks of length 128 bits (SK_i), where i=1,2,3,.....

1. Convert the Seed Key (SK) into Seed Key Matrix (SKM).
2. Compute the Main Matrix (MM) as follows:
   \[ \text{MM}_i = \text{SKM} \text{ xor } \text{IM} \] (each bit in the matrix with its correspondence in other matrix).
3. Perform the MM permutation on MM_i.
4. Perform the IM permutation on IM_i.
5. Compute the Horizontal Shift Matrix on MM_i.
6. Compute the Vertical Shift Matrix on MM.
7. Perform the combining step by XORing the last MM with the matrix from IM.
8. Convert Block to Stream and output key.
9. If not end of stream inputs go to 2.
10. Stop

Figure 3  The Proposed Algorithm
4-3-1. Seed Key

The seed key is the secret key of the algorithm, its consists of 16 characters that converted into 112 bits (7 bits for each character using ASCII table), the next step is distributing the 112 bit in a matrix row by row (eight bits in each column), the last raw will be calculate it by adding the Rows (begin from row1 to row 7) and put the result in row 8 which then reformed as an 8*16 Seed Key Matrix (SKM).

4-3-2. IM Permutation

A fixed table of permutation generated randomly once (as described in figure 4) and used to permutated the Initial Matrix IM, the main task for this operation is to rearrange the bits in a matrix form as a nonlinear function performed on the initial matrix.

To explain how this operation can be performed the number 69 in the permutation matrix means rearrange the position of the bit in the position 69 to be in the position number 1, and next the number 82 means put the bit in the position 82 in position number 2, and so on.

<table>
<thead>
<tr>
<th>69</th>
<th>82</th>
<th>105</th>
<th>127</th>
<th>76</th>
<th>2</th>
<th>56</th>
<th>40</th>
<th>80</th>
<th>98</th>
<th>56</th>
<th>22</th>
<th>55</th>
<th>20</th>
<th>29</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>3</td>
<td>96</td>
<td>103</td>
<td>128</td>
<td>50</td>
<td>106</td>
<td>76</td>
<td>4</td>
<td>71</td>
<td>6</td>
<td>45</td>
<td>70</td>
<td>27</td>
<td>81</td>
<td>10</td>
</tr>
<tr>
<td>104</td>
<td>124</td>
<td>5</td>
<td>73</td>
<td>107</td>
<td>105</td>
<td>72</td>
<td>49</td>
<td>114</td>
<td>57</td>
<td>117</td>
<td>90</td>
<td>25</td>
<td>8</td>
<td>116</td>
<td>120</td>
</tr>
<tr>
<td>75</td>
<td>16</td>
<td>74</td>
<td>7</td>
<td>77</td>
<td>86</td>
<td>104</td>
<td>108</td>
<td>58</td>
<td>72</td>
<td>121</td>
<td>23</td>
<td>121</td>
<td>1</td>
<td>59</td>
<td>44</td>
</tr>
<tr>
<td>42</td>
<td>125</td>
<td>113</td>
<td>94</td>
<td>9</td>
<td>62</td>
<td>18</td>
<td>48</td>
<td>91</td>
<td>109</td>
<td>21</td>
<td>89</td>
<td>68</td>
<td>45</td>
<td>93</td>
<td>24</td>
</tr>
<tr>
<td>83</td>
<td>95</td>
<td>51</td>
<td>126</td>
<td>78</td>
<td>11</td>
<td>65</td>
<td>87</td>
<td>14</td>
<td>19</td>
<td>28</td>
<td>46</td>
<td>110</td>
<td>99</td>
<td>115</td>
<td>119</td>
</tr>
<tr>
<td>64</td>
<td>97</td>
<td>84</td>
<td>102</td>
<td>54</td>
<td>114</td>
<td>13</td>
<td>92</td>
<td>17</td>
<td>61</td>
<td>47</td>
<td>75</td>
<td>101</td>
<td>118</td>
<td>68</td>
<td>26</td>
</tr>
<tr>
<td>41</td>
<td>52</td>
<td>85</td>
<td>53</td>
<td>79</td>
<td>66</td>
<td>112</td>
<td>15</td>
<td>111</td>
<td>67</td>
<td>123</td>
<td>88</td>
<td>60</td>
<td>112</td>
<td>100</td>
<td>43</td>
</tr>
</tbody>
</table>

Figure 4  IM Permutation matrix

4-3-3. Main Matrix Permutation

In this step the main matrix (MM) that is produced from the initialization step is permutated (according to the permutation table explained in figure 5) to generate a huge diffusion for the output block and performed as explained in the previous section.
In order to spread the effect of each single bit to others, the position of the bits must be changed for a long period and the permutation function have no calculated period, therefore, another position changing function needed to guaranteed that bits positions are changed periodically, this function besides the permutation function must produce a long period. To generate such long period a Linear Feedback Shift Register (LFSR) is candidate, each row in the matrix can be considered as a single LFSR of length 16 and the tapped stages are (16,3) to generate a maximum period equal to \((2^{16}-1)\), this operation called horizontal shift in the general structure (see figure 6), in the same time each column in the matrix can be considered as a single LFSR of length 8 and the tapped stages are (8,5) to generate a maximum period equal to \((2^{8}-1)\), this operation called vertical shift in the general structure(see figure 7), so the total period will be \((2^{16}-1)\times (2^{8}-1)\approx 2^{24}\). The operation of horizontal shift must perform first on the whole matrix and then the vertical shift operation. In fact, the period must be greater than \(2^{24}\) because it must be multiplied by the permutation period which must be calculated manually.

The period generated by the matrix shifting is a factor of the whole system period multiplied by other factors from the combining function and the permutation functions and it is a period of the block diffusion pictures.

4-3-5 Matrix shifting

![Figure 5 MM Permutation](image)

![Figure 6 Horizontal shift steps](image)
4-3-6 Combining Function

The combining function is performed by recomputing the Main Matrix by XORing each bit from the Main Matrix (MM) with its corresponding bit of the IM. The output stream is generated by converting the resulted MM into stream of length 128 bits.

5. Complexity Measurement

To calculate the complexity of the algorithm we need to specify the period of the key generator algorithm which will be equivalent to the complexity of the system[6]. The period of the algorithm can be computed by $2^{112}$ (all possible seed key) * $2^{128}$ (all possible IM's)*128! (all possible IM permutation)*128! (all possible IM permutation), which approximately equal to $(2^{1040})$ and exceeds any input sequence in an applicable algorithm and might not be attacked in a brute-force attack. The algorithm cannot be attacked by any other known attacking method because of the high computational complexity of the permutation and matrix shifting.

6. Randomness Tests

The proposed algorithm is stream cipher algorithm with a fashion of block cipher algorithm, therefore, it has a block cipher algorithm features. One of these features is the output stream is not a continuous stream of bits but a sequence of blocks.

The statistical tests of randomness are a stream cipher test or can be performed on a continuous stream of bits; therefore, it will never work with block cipher algorithm[6].
A single block can be tested using the statistical tests of randomness to check its diffusion. In other words, the block has a high diffusion if it passes the statistical tests of randomness, Figure 8 show three tests results.

7. Conclusion and recommendations

We can conclude that the proposed algorithm:-

1. is a secure high speed stream cipher algorithm, it must need to prove that it has a high complexity, long periodicity, high nonlinearity and must pass the randomness tests.

2. The period of the algorithm can be computed by $2^{112}$ (all possible seed key) * $2^{128}$ (all possible IM's)*128! (All possible IM permutation)*128! (all possible IM permutation), which approximately equal to $(2^{1040})$ and exceeds any input sequence in an applicable algorithm and might not be attacked in a brute-force attack. The algorithm cannot be attacked by any other known attacking method.

3. The periodicity of the algorithm obtained will exceed the length of any input media might be encrypted by any other algorithm.

And the recommendation is that if the algorithm implement on parallel processor hardware, the result is better, it means the speed is faster then the speed on a single processor.

<table>
<thead>
<tr>
<th>Three keys</th>
<th>Frequency Test</th>
<th>Serial Test</th>
<th>Poker Test</th>
<th>Run Test</th>
<th>Autocorrelation Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEY 1</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
</tr>
<tr>
<td>KEY 2</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
</tr>
<tr>
<td>KEY 3</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
</tr>
</tbody>
</table>

Figure 8 The result of tests
8. References

[1] Ala. G konheim (Computer security & cryptography) 2007 Wil-
intersection a john wily & sons,inc,pupliction.
Springer 2010
London 2007
of the Hermes 8-128 Stream Cipher", School of Science and Technology
in Hellenic Open University, Patras, Greece.
[7]: Niels Ferguson, Doug Whiting, Bruce Schneier, John Kelsey, Stefan
Lucks, and Tadayoshi Kohno, " Fast Encryption and Authentication in a
Single Cryptographic Primitive", University at Mannheim.