Compact High Speed Reconfigurable Hardware Implementation of RC4 Stream Cipher

Priya Nagar¹, N.B. Hulle²

(Department of E&TC, PG student G.H. Raisoni Institute of Engineering and Technology, Wagholi, Pune University, Pune)

Abstract: RC4 Stream cipher is well known for its simplicity and ease to develop in software. But here, in the proposed design we have highlighted the modified hardware implementation of RC4. As RC4 is the most popular stream cipher, the proposed design performs reading and swapping simultaneously in one clock cycle. The proposed design also highlights the adder part which enhances the architecture speed. As this design uses fast Carry Look Ahead Adder as the adder logic. RC4 uses a variable length key from 1 to 256 bytes to initialize a 256-byte array. The array is used for subsequent generation of pseudo-random bytes and then generates a pseudorandom stream, which is XORed with the plaintext/ciphertext to give the ciphertext/plaintext. The RC4 stream cipher works in two phases. The key setup phase and the pseudorandom key stream generator phase. Both phases must be performed for every new key. The RC4 algorithm will be implemented by FPGA using VHDL software platform.

Keywords: CLA, Clock, FPGA, KRAM, RC4, SRAM, Throughput

I. Introduction

Message secrecy is one of the most important aspects of communication but especially in wireless environment messages are highly insecure and encryption is must in such environment. The various encryption algorithms are available but RC4 encryption algorithm is stream type and can be implemented in hardware and software. The RC4 stream cipher is implemented in hardware by Sourav Sen Gupta¹, Koushik Sinha², Subhamoy Maitra¹, and Bhabani P. Sinha¹ Indian Statistical Institute, 203 B T Road, Kolkata 700 108, India 2 Honeywell Technology Solutions Lab, Bangalore 560 076, India and also by P. Kitsos, G. Kostopoulos, N. Sklavos, and O. Koufopavlou VLSI Design Laboratory, Electrical and Computer Engineering Department, University of Patras, Patras, Greece. This hardware implementation is fast and reliable as compared to software implementations and block ciphers algorithms. Here is the basic concept of RC4. RC4 is a stream cipher and it can cipher individual units as they occur. It can cipher individual data elements immediately, as they arrive. RC4 is a stream cipher signature, and can be identified by analysis of the design. So it takes less time to generate the cipher text. RC4 algorithm uses stream cipher that is often used in application where plaintext comes in quantities of unknown length. RC4 does not require block filling, so does not need block padding, and does not need a padding removal structure. A particular RC4 algorithm key can be used only once. As RC4 is an stream cipher which makes encryption faster than the other algorithms that uses block cipher. The chance of losing the data in wireless transmission is very high, but RC4 algorithm can easily synchronize with the transmission even if the data is lost. That is why RC4 is widely used in wireless networks. RC4 algorithm is implemented in software, so the complexity is reduced & it is cheaper as the software can be easily changed according to the requirements.

II. Methodology

Stream cipher as shown in Fig. 1, in which the first block indicates encryption phase and second block indicates decryption phase. The transmitter encrypts plain text with key stream, which is generated by the key stream generator with the private key i.e. distributed on secure channel. On the other hand, the receiver decrypts the received cipher text with key stream which is generated by the key stream generator with the private key i.e. also distributed on secure channel.
Fig 1. Block diagram of Stream Cipher

Fig 2 shows the block diagram of proposed modified RC4 algorithm which uses a variable key length from 1 to 256 bytes to initialize a 256 byte array. The array is used for subsequent generation of pseudo-random bytes and then generates a pseudorandom stream, which is XORed with the plaintext / cipher text to give the cipher text / plaintext.

Fig 2. Block diagram of modified RC4

There are 256-byte arrays, S-Box and K-Box. The S-array is filled linearly such as S0=0, S1=1, S2=2………..S256=256. The K-array consists of the key, repeating as necessary times, in order to fill the array. The RC4 stream cipher works in two phases. The key setup phase and pseudorandom key stream generator phase. Both phases must be performed for every new key.

Key setup phase:
For i = 0 to 127
J = (j + S(i) + K(i) )
Swap S (i) and S (j).

Fig 3(a). Key Setup Phase

RC4 uses two counters, i and j, which are initialized to zero. In the key setup phase the S-box is being modified according to pseudo-code: once the Key Setup is completed the second phase encrypts or decrypts a message. The pseudorandom number generator (PRGN) phase is described by the following pseudo code:

Key stream generator phase:
i = i+1;
j = j+S (i)
Swap S (i) and S (j)
t = (S (i) +S (j))
K = S (t)

Fig 3(b). Keystream generation Phase

The key stream K is XORed with the plaintext / cipher text to produce cipher text / plaintext. Recent improvements make FPGAs increasingly suitable for cryptanalysis due to high density and high on chip memory bandwidth. The proposed design is more effective as it uses less number of components than the existing system[1]and [2]. This makes it more efficient and cheaper.
In our proposed design swapping and reading is done simultaneously in a single clock cycle. While in the existing design [1] swapping and reading are two separate tasks. The proposed design performs swapping and reading simultaneously by using the two main concepts i.e. Loop Unrolling and hardware Pipelining while the existing design [1] uses only the loop unrolling method. The already existing design [1] has used normal adders while in the proposed design we have used fast Carry Look Ahead adders which not only optimizes the area but also helps in increasing the encryption speed. Fast CLA is a new concept implemented in this paper. Here Carry Look Ahead is implemented by using NAND gate instead of AND gate.

**Fig4. Algorithm for RC4 Phases**

Device utilization summary:

<table>
<thead>
<tr>
<th>Component</th>
<th>Utilized</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Slice Registers</td>
<td>4144</td>
<td>1%</td>
</tr>
<tr>
<td>Number of Slice LUTs</td>
<td>5689</td>
<td>2%</td>
</tr>
<tr>
<td>Number of fully used Bit Slices</td>
<td>2101</td>
<td>27%</td>
</tr>
<tr>
<td>Number of Bonded IOBs</td>
<td>143</td>
<td>14%</td>
</tr>
<tr>
<td>Number of BUFG</td>
<td>2</td>
<td>6%</td>
</tr>
</tbody>
</table>

Selected Device : 5vlx330tf1738-2

**IV. Conclusions**

In this paper hardware implementation of RC4 performing swapping and reading simultaneously in one clock cycle is highlighted. The proposed design provides better result in terms of number of components used, type of adder, methods of computation and number of clock cycle. The proposed design uses fewer components, which is major advantage of this system. These ciphers were coded in VHDL language and synthesized in an FPGA device.

**References**

[1]. One Byte per Clock: A Novel RC4 Hardware Sourav Sen Gupta1, Koushik Sinha2, Subhamoy Maitra1, and Bhabani P. Sinha1 1 Indian Statistical Institute, 203 B T Road, Kolkata 700 108, India 2 Honeywell Technology Solutions Lab, Bangalore 560 076, India.


[3]. The Fastest Carry Lookahead Adder Yu-Ting Pai And Yu-Kung Chen Department of Electronic Engineering Huafan University .


DOI: 10.9790/4200-05125456 www.iosrjournals.org 56 | Page