ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 3, Special Issue, March-April 2013, An ISO 9001: 2008 Certified Journal.
E-NSPIRE, A National Level Conference held at Pravara Rural Engineering College, Loni, Maharashtra, INDIA.

# Hardware Implementation of RC4 Stream Cipher using VLSI

**D. B. Rane, Swetal R.Gund, Chandreshwari B. Pawar, Jyoti F. Ukande**

**Abstract: This paper deals with the design of RC4 stream cipher for wireless LAN Security. 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 EXOR ed with the plaintext/cipher text to give the cipher text/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. Previous designs supports 1044 bytes storage locations while this design uses only one 512 bytes storage locations. The system is implemented in hardware by using VHDL language and Xilinx FPGA device. RC4 is the most widely used software based stream cipher. The cipher has been integrated into TLS/SSL and WEP implementations. The cipher was design by Ron Rivest in 1987 and kept as a trade secret until it was leaked out in 1994. RC4 is extremely fast and its design is simple. This paper deals with RC4 keystream generator, within the scope of the existing model of an exchange shuffle, in order to achieve better security. The main factors in RC4's success over such a wide range of applications are its speed and simplicity: efficient implementations in both software and hardware are very easy to develop.**

***Keywords*** - **RC4, Stream cipher, KRAM, SRAM, Data Serializer, K Stream Serializer.**

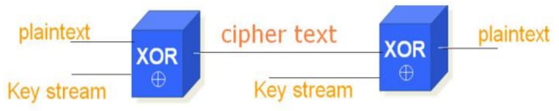
## I. INTRODUCTION

Cryptography is a Greek word for "hidden writing. "The art and science of transforming (encrypting) information (plaintext) into an intermediate form (cipher text) which secures information in storage or transit. Normally, security occurs as a result of having a vast number of different transformations. Then, if an opponent acquires some cipher text, a vast number of different plaintext messages presumably could have produced that exact same cipher text, one for each of the possible keys. Cryptography is a part of cryptology, and is further divided into secret codes versus ciphers. As opposed to steganography, which seeks to hide the existence of a message, cryptography seeks to render a message unintelligible even when the message is completely exposed. Cryptography includes at least: Key generation, Secrecy, Message authentication (integrity). Wireless Local Area Network (WLAN) technology shows bright future for different wireless communications industries. The IEEE 802.11 is one of the most widely used wireless standards. IEEE 802.11 standard does not provide implementation details, but provides specifications for physical and Media Access Control (MAC) layers. This standard is widely used in ad-hoc and client/server networks but special attention should be given to the security of data in transmission channel. The security of this standard is based on Wired Equivalent Privacy (WEP) scheme. Currently, all the IEEE 802.11 products support WEP, but IEEE 802.11i working group introduced the Advanced Encryption Standard (AES), as the security scheme for the future IEEE 802.11applications. The WEP uses two basic components, Pseudorandom Number Generation (PRNG) and the integrity algorithm. The PRNG is the most important component because it is the actually original encryption core. WEP adopts RC4 stream cipher as PRNG unit and Cyclic Redundancy Check (CRC-32) as the integrity algorithm. WEP has several limitations and encryption procedure has no provision for key rotation, users have to transmit the data using same single key; which made cracking of WEP even easier. In response to growing wireless security importance in corporate field, the Wireless Fidelity (Wi-Fi) alliance proposed a new security protocol Wi-Fi Protected Access (WPA). WPA uses RC4 stream cipher as a security algorithm with new dynamic key management method, known as Temporal Key Integrity Protocol (TKIP). TKIP uses Message Integrity Code (MIC) instead of WEP’s CRC-32 for ensuring data integrity. WPA is a secure solution for upgradable equipment of WEP but not supporting to WPA2. As far as transmission speed is concerned WPA has edge over WPA2.

## II. RC4 STREAM CIPHER

RC4 uses a variable length key from 1 byte to 256 bytes to initialize a 256-byte array. There are two 256-bytes arrays, S-Box and K-Box. The S-array is filled linearly such as S0=0, S1=1, S2=2 ……….. S255=255. The K-array consists of the key, repeating as necessary times, in order to fill the array. The RC4 key is often limited to 40 bits, because of export restrictions but it is sometimes used as a 128 bits key. It has the capability of using keys between 1 byte and 256 bytes. The RC4 algorithm works in two phases, key setup and pseudorandom key stream generation phase. Key setup is the first and most difficult phase of this algorithm. During N-bit key setup (N being your key length), the encryption key is used to generate an encrypting variable using two arrays, sarray, k-array and N-number of mixing operations. These mixing operations consist of swapping bytes according to RC4 algorithm. Figure 1 shows the block diagram of the RC4 two phases.

ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 3, Special Issue, March-April 2013, An ISO 9001: 2008 Certified Journal.
E-NSPIRE, A National Level Conference held at Pravara Rural Engineering College, Loni, Maharashtra, INDIA.



RC4 uses two counters, counter i and counter j, which are initialized to zero value. In the key setup phase the S-box is being modified according to pseudo-code:

***Key setup phase:***

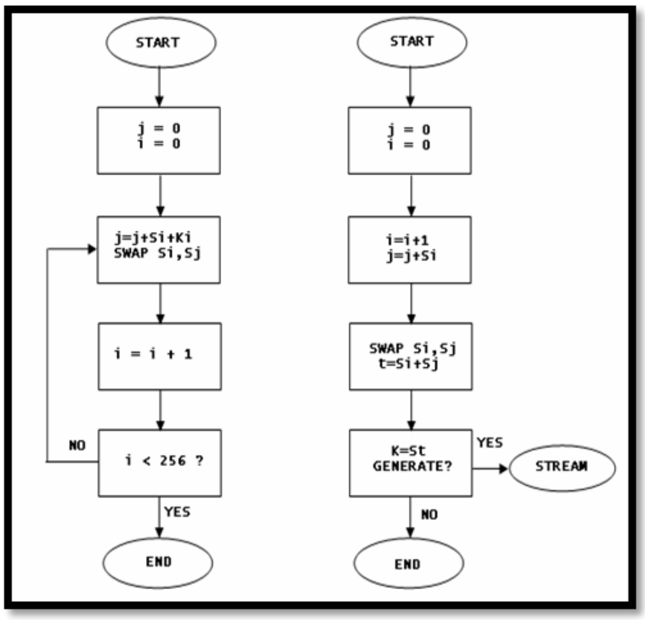
```
For i= 0 to 255
j = (j + Si + Ki) mod 256
 Swap Si and Sj
```

Once the encrypting variables are produced from the key setup phase, it enters in the pseudorandom key stream generation phase. The pseudorandom key stream generation phase is given by the following pseudo code:

***Key stream generation phase:***

```
i=(i+ 1)mod256
j = (j + Si) mod 256
 Swap Si, and Sj
t = (Si + Sj) mod 256
K=Si
```

Where this generated pseudo random code is XORed with the plain text/cipher text to create cipher text/plain text. XOR is the logical operation of comparing two binary bits. If the bits are different, the result is 1. If the bits are the same, the result is 0. Once the receiver gets the encrypted message, he decrypts it by XORing the encrypted message with the same encrypting variable.

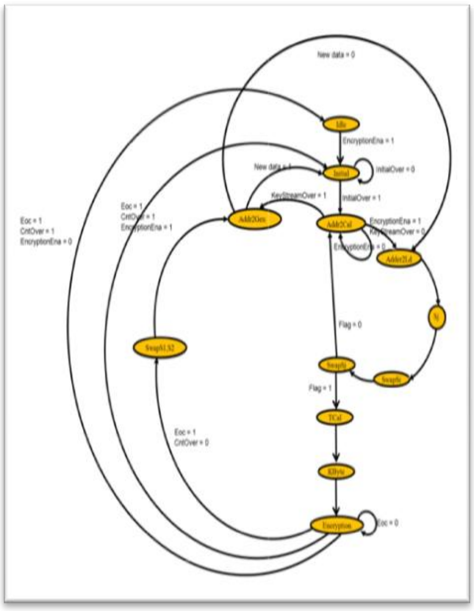


## III. PROPOSED ARCHITECTURE

The implementation of the storage unit S-array andk-array is shown in Fig. 2. The S-box RAM consists of three256 bytes RAM blocks. S-box RAM has four inputs and one output. The two inputs are the read and write signals while the other two are the address and data signals. Also, the RAM box has the signals of clock and reset. The operation of RAM blocks is quite simple. If the reset signal occurs the blocks are initialized linearly. For Ram block, if the write signal occurs new data are stored in the address position, or if the read signal occurs the data in the address position are available on the output of the block .Figure 3 shows the state diagram for proposed architecture. It goes through Initial state, Adder2Cal state, Adder2Ld state, Sj state, Swap Si state, Swap Sj state, Addr2Gen state, TCal state, Kbyte state and Encryption state to complete key setup generation phase and key stream generation phase .Figure 4 shows hardware module of proposed architecture, which consists of two storage units, S-RAM and K-RAM. It also consists of Addr1 and Addr2. The function of addr1 is to generate initial data and address for S-Ram. The function of addr2 is to generate address for K-Ram. The Adder1 is used to calculate j = j+ Si + Ki and Sreg1, Sreg2, with Data Mux and Data D-Mux are used for swapping the contents of S-Ram block. The hart of architecture is SMC which generates control signals for all modules. The Keystream serialize and Data serialize are used for parallel to serial conversion and then EX-ORing is performed to generate encrypted signal. Initial State: In this state we first fill the SRAM and KRAM. To fill both the RAM, the Data is given directly to the KRAM for filling the Data randomly as a Data Bus and address is given by Addr1. The Data at SRAM is filled with the help of Addr1 which gives the address and at same time same data comes from the Data Mux and are loaded at same address. Addr1 gives the location from 0 to 255 (as cnt255) in both the RAM. The output of Addr1 is given to the KRAM that gives the address for Data Bus of KRAM. The input of SRAM (as Addr[7:0]) shows address and Data fill linearly

From Data In as S0=0, S1=1, S2=2, …………S255=255. The initial state exists till the SRAM / KRAM fill completely (all255 location). After filling, Initial Over = 1, and state go toAddr2cal state.Addr2cal state. In this state, Mux gives the Key Data Out as Mux Out by selecting the Sel=1, and SRAM gives the data through Data Out to Data DMux. And Sel Data Out select this Data Out as SR1 and load this value inS_Reg1. All these values from A2, SR1 and Mux Out are added. Adder2Ld state:In this state the output of Adder1(Adder1Out) is loaded into the Addr2 register. SJ state loaded value at Addr2 gives j value. This j value is the address, which is selected by SelAddr of Addr Mux. The output of Addr Mux gives address of SRAM. The value of that location is obtained as Data Out, selected by Data DMux as SR2 and stored at S_Reg2 as SJ Swap SI**:** In this state, the address is taken from Addr2, which is selected by Addr Mux as Addr and the data is loaded on this address of SRAM as Data In by selecting Reg2 through Data Mux. Swap SJ**:** In this state, the address is taken from Addr1, which is selected by Addr Mux as Addr and the data is loaded on this address of SRAM as Data In by selecting Reg1 through Data Mux. All process of like Addr2cal, Addr2ld, SJ, Swap SI and Swap SJ occurred 255 times. When Keysetupover=1(high) and Flag=1(high) then it goes to Addr2Gen state. Addr2 Gen state**:** At this state again we obtain the value of J by Key Setup phase .

For i = 0 to 255
j = [j+Si+Ki] ,
Here, Mux out is taken 00000000, so
j = j+Si and i=i+1;
In this state, Mux In (00000000) is selected as output of Mux, and the whole process of Addr2Cal state is repeated. After this it follows the process of Addr2 LD SJ, Swap SI and Swap SJ. These processes occur just once. **TCal state:** When Swap SJ complete, flag will high and reaches the TCal state. At TCal state we perform the following operation. t = Si +Sj So in this state the value of S_Reg1 and S_Reg2 are added at Adder2 register. The output of Adder2 i.e. Adder2Out value goes to the Addr Mux. The output of Addr Mux is selected by selAddr as Adder2Out. This output of Addr Mux gives the address and the Data at this address comes out through Data DMux as KeyStream Out. **KByte state:** The data that was given to FIFO in the Initial state is now loaded into Data Serializer. At the same time the data from Data DMux is loaded into K_Stream Serializer. **Encryption state:** The key data byte from K_Stream Serializer and plaintext from Data Serializer comes out serially in the form of bits, are XORed to gives the output as encrypted data. EOC becomes high whenever a byte of plaintext is encrypted.



**State Diagram of Proposed Architecture**

## IV. DESCRIPTION OF BLOCK DIAGRAM

### *A.S-RAM & K-RAM*

Proposed architecture consist of 256 byte RAM and this RAM used as S-RAM1 (S1), S-RAM2 (S2), K-RAM1 (K1) and K-RAM2 (K2) with necessary control unit and addressed generator logic. The implementation of the storage unit S-array and k-array is shown in figure. The S-box RAM consists of three 256 bytes RAM blocks. S-box RAM has four inputs and one output. The two inputs are the read and write signals while the other two are the address and data signals. Also, the RAM box has the signals of clock and reset. The operation of RAM blocks is quite simple. If the reset signal occurs the blocks are initialized linearly. For Ram block, if the write signal occurs new data are stored in the address position, or if the read signal occurs the data in the address position are available on the output of the block [01, 02, 03]. Due to parallel structure of this architecture two parallel streams are generated from S-RAM1 and SRAM2.

### *B. Multiplexer*

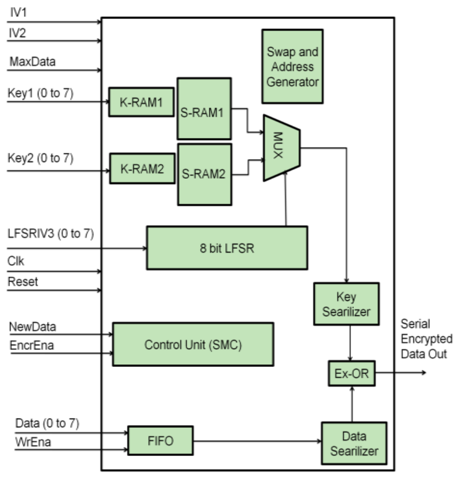
Here used 2:1 mux. More confusion at the output side can be added by selecting one of the streams randomly from MUX out by using 8 bit LFSR as select lines, whose feedback function is represented by primitive polynomial x8 + x4 + x3 + x2 + 1. The MUX output is pseudorandom byte converted into serial form using key searilizer.

### *C. Key searillizer& Data searillizer*

The MUX output is pseudorandom byte converted into serial form using key searilizer. The eight byte FIFO is used to store eight bytes of data and converted into serial form by data searilizer. Finally two streams are Ex-ORed to obtain encrypted serial data out .

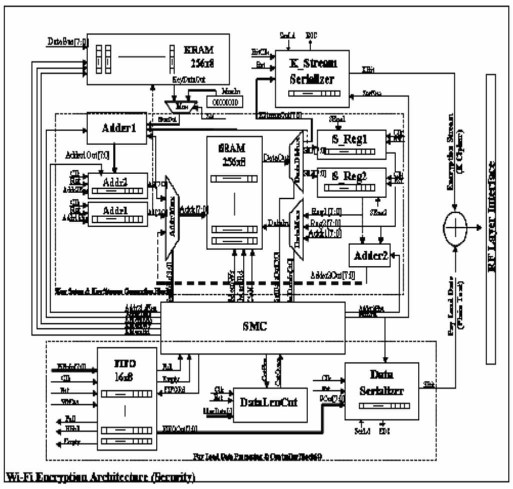
### *D. Linear Feedback Shift Registers (LFSR)*

An LFSR consists of clocked storage elements (*flip-flops*) and a *feedback path*. The number of storage elements gives us the *degree* of the LFSR. In other words, an LFSR with *m* flip-flops is said to be of degree *m*. The feedback network computes the input for the last flip-flop as XOR-sum of certain flip-flops in the shift register. As we have learned so far, stream ciphers use a stream of key bits *s*1, *s*2, . . .that are generated by the key stream generator, which should have certain properties. An elegant way of realizing long pseudorandom sequences is to use linear feedback shift registers (LFSRs). LFSRs are easily implemented in hardware and many, but certainly not all, stream ciphers make use of LFSRs. As we will see, even though a plain LFSR produces a sequence with good statistical properties, it is cryptographically weak. LFSRs can make secure stream ciphers. It should be stressed that there are many ways for constructing stream ciphers.



**Block Diagram of Proposed Architecture of RC4 Stream Cipher**

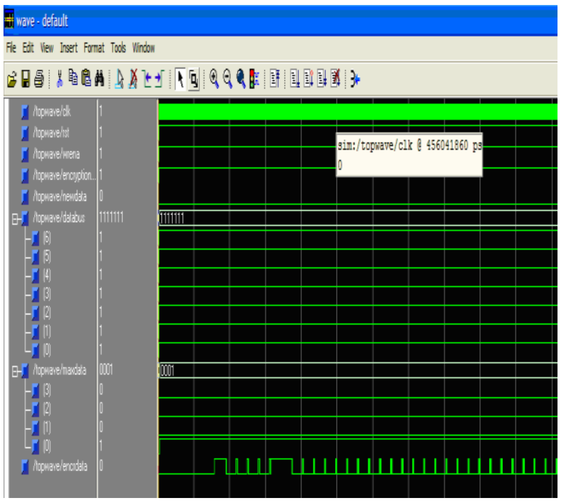
ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 3, Special Issue, March-April 2013, An ISO 9001: 2008 Certified Journal.
E-NSPIRE, A National Level Conference held at Pravara Rural Engineering College, Loni, Maharashtra, INDIA.



Hardware module

## V. VLSI IMPLEMENTATION RESULTS

The proposed architecture was captured by using VHDL. All the system components were described with structural architecture. The system tested using confirmed test vectors in order to examine its correctness and simulated by Model sim simulator, Figure 5 shows output waveforms for the same. The whole design was synthesized, placed and routed by using XILINX FPGA device.



Output Wv

## VI. CONCLUSION

A hardware implementation of the RC4 stream cipher for *wireless LAN Security* is presented in this paper. The proposed design provides high data throughput using variable key length from 1 byte to 16 bytes. It provides high flexibility as it can be used in many applications with any key length from 1 byte to 256 bytes. The proposed system achieves a data throughput up to 22 Mbytes/sec in a clock frequency of 64 Mhz. This design requires only 512 bytes storage, while previous designs supports for 1044 bytes storage. The comparison of these results with previous results proves that this new reduced hardware implementation is better solution for any cryptographic system.

## REFERENCES

[1] Stephen Brown Zvonk Vransic, Fundamentals of Digital Logic Design with VHDL, Second editation, Tata McGraw Hill, 2005.

[2] Douglas A. Pucknell, Kamran Eshraghian, Basic VLSI design, 3rd Edition, Prentice Hall of India, 2004.

[3] Morris Mano, Digital Design, Third editation, Prentice Hall of India, 2006.

[4] Andrew S. Tanenbaum, Computer Networks, Second editation, Prentice Hall of India, 1991.

[5] Kostopoulos, N. Sklavos and O.Koufopavlou.VLSI design laboratory.

### Author's Profile

**Prof. D. B. Rane**
**dbrane_123@rediffmail.com**
M.E. Electronics Engg (Digital Electronics)
Asst. Prof. , Electronics Engg**.**
Pravara Rural Engineering College, Loni,

**Shwetal R. Gund**,
s.r.gund@gmail.com.
B.E. Electronics Engg. (Persuing)
Pravara Rural Engineering College, Loni,

**Chandreshwari B. Pawar**,
shwaripawar385@gmail.com.
B.E. Electronics Engg. (Persuing)
Pravara Rural Engineering College, Loni,

**Jyoti F. Ukande**,
jfukande12@gmail.com.
B.E. Electronics Engg. (Persuing)
Pravara Rural Engineering College, Loni,