Synthesis and Simulation of FPGA Based Rc4 Encryption Method

Renu M. Sharma[1] (*M.Tech 2nd year)*
ECE Department
Shri Balaji Institute of Tech & Management
Betul (M.P.), India
Email: renusharma1123@gmail.com[1]

Pallavi Choudhary[2] *Asst. Professor*
ECE Department
Shri Balaji Institute of Tech & Management
Betul (M.P.), India
Email: pallavichoudhary18@gmail.com[2]

***Abstract*—**This electronic document is a "live" template. The various components of your paper [title, text, heads, etc.] are already defined on the style sheet, as illustrated by the portions given in this document. DO NOT USE SPECIAL CHARACTERS, SYMBOLS, OR MATH IN YOUR TITLE OR ABSTRACT.

Keywords-*component; formatting; style; styling; insert (key words)*

*****

I. INTRODUCTION

Cryptography is a method of storing and transmitting data in a safe/secured way/fashion which can be read and process it for whom it is send. Cryptography is a Greek word and the meaning of this word describes the "hidden writing." The cryptography is a technology which transforms the information into an intermediate form to secure the information. The cryptography mainly works on the problems, which are associated with authentication, secrecy and integrity. Cryptography also related with the meaning of protocol. A sequence of actions is known as protocol, which are concern two or more sides, designed to fulfill a goal. Thus, the cryptography is used by the cryptographic protocol. A cryptographic algorithm is used by this protocol and its intention is to prevent attempts of thefts and invasions. Now a days using terms of Theory of Statistics and Theory of numbers cryptography has been developed in a science. Many kinds of cryptographic algorithms have been invented in order to handle all the cryptographic problems. There are several categories of cryptographic algorithm based on the complexity of these problems. A much known and useful cryptographic algorithm for solving cryptographic problems is the RC4 stream cipher. Normally, a vast number of different transformations occur in the security. If an opponent gets some cipher text, a various number of different plaintext messages presumably could have produced that exact same cipher text, one related for every of the possible keys. Cryptography is a section of cryptology. It is further divided into secret codes versus ciphers.Cryptography attempt to render a message which is impossible to understand or unintelligible even when the message is completely exposed. Where as Steganography seek to hide the existence of a message. Cryptography includes at least: Key generation, Secrecy, Message authentication. RC4 is a cipher stream which was designed by Ron Rivest Cipher 4". The trade secret behind RC4 was revealed in September 1987 of RSA Security; while it is then termed as "Rivest Cipher 4". The trade secret behind RC4 was get exposed to others when the description of the cipher was sent to the Cypherpunks mailing list (which is an group of people interested in privacy and cryptography, for communication through mailing). After that, the description was posted on many website and the genuineness of the information was confirmed. The outputs result of these described cipher were matching with the outputs of licensed RC4. RC4 attained the apex popularity because of its simplicity and efficiency. Because the algorithm is known, it doesn't remain a trade secret. It was used in many commonly used standards protocols such as WEP, WPA, SSL or TLS. Its impressive speed and simplicity are the main factors due to which it is deployed over such a wide range of applications. There are two phases in which RC4 stream cipher works. The key setup phase and pseudorandom key stream generation phase. Both phases must be performed for every new key [01]. This RC4 stream cipher supports variable key lengths from 1 byte to 256 bytes. It uses only one 256 bytes S-array and one 256 bytes K- array. In this algorithm for key setup phase three clock cycles are required for each byte generation and the pseudorandom key stream generation phase three clock cycles are required per byte generation. In the two ways i.e. either hardware or software any algorithm can be implemented. On many factors such as algorithm performance, cost and flexibility the choice of platform depends. The hardware always appears to be the best choice for secure high speed networks because hardware implementation of cryptographic algorithms has high security level and run faster than software. In hardware implementations, The FPGAs are the suitable platform for cryptographic applications because of the flexibility and high speed capability. The FPGA is reconfigurable means that depending on security and application requirements it can be re-programmed to perform the more intensive operations of am range of ciphers. In this paper simulation and synthesis results of RC4 stream cipher is presented. In the modern information age, the notion of security often confirms to the idea of confidentiality of information; especially of digital information utilized and broadcast over various communication networks. It is important to note that the requirement for confidentiality and security is in fact entirely a social construct. In fact, the security is still judged qualitatively in terms of the eagerness and competence of an adversary.

Cryptology – Inherently a social science refers to the art of bridging this bizarre gap between an entirely social notions called 'security' and the logical foundations of mathematics, Computer Technology /science and associated domains. After some research on the web to find an interesting cryptographic primitive to implement, it has been decided to implement RC4. There are several reasons for choosing this stream cipher. First of all, this cipher is one of the most commonly used stream

cipher. Moreover it is used by really important and famous protocols and standards such as SSL, TSL, WEP, etc. Another reason for this choice is that it is well known for its simplicity and efficiency.

This paper is organized as follows. In Section 2 description of algorithm is explained. In these section types of stream cipher is explained. Section 3 explains the RC4 algorithm implementation i.e. key scheduling algorithm and pseudorandom number generation algorithm. Section 4 describes the steps of the RC4 algorithm. Section 5 shows the synthesis and simulation results of RC4 algorithm for key length of 256-bits and 56-bits. Finally, conclude this paper in Section 5.

II. DESCRIPTION OF RC4 ALGORITHM

RC4 is one of the most widely used ciphers because of its simplicity and fast speed. It is also used as default cipher in Internet protocol like SSL (Secure Socket Layer) to protect Internet traffic and WEP to secure wireless networks. RC4 is also called ARC4 or ARCFOUR. RC4 can be implemented in software and hardware. In software implementation it provides fast speed but there is more possibility of hacking the data in software as far as hardware is concerned. The data security is more as compared to software using RC4 cipher in hardware implementation. RC4 cipher has some weaknesses which make it less selective cipher for new systems. It is especially Vulnerable i.e. its security level is less when some of the beginning keys of the output key stream are not discarded, non- random or related keys are used, or a single key stream is used twice. To increase the security level of RC4 cipher it is required to discard some starting keys which increase its randomness. It is required to avoid selection of related keys. RC4 Algorithm consists of two algorithms i.e. KSA (Key Scheduling Algorithm) and PRGA (Pseudorandom Generation Algorithm). The sequence of pseudorandom for 8-bit security keys are generated by using these two algorithms. These are called as pseudorandom security keys because these keys fulfill the properties of random numbers. These 8-bit security keys are XORed with the data i.e plaintext. After XORing the 8-bit security keys with plaintext the new data is obtained which is called as Cipher text. This cipher text is then transmitted through the media which can be wired or wireless. At the receiver end the received cipher text is again XORed with the sequence of the keys which had been generated at the transmitter end. It means that RC4 algorithm is required to execute at both the transmitter and receiver end to generate same security 8-bit keys sequence. If the generation of the sequence security key generation at the transmitter and receiver end is not same then correct data cannot be recover at the receiver side. The key stream is generated from a variable length key using an internal state consists of the following elements: i) A 256 bytes S-array containing a permutation of these 256 bytes. ii) Two indexes i and j, used to point elements in the S array. The S-Array is first initialized and then shuffled. For the shuffling of S-Array the Key Scheduling Algorithm (KSA) is executed. As soon as the S array has been initialized and "shuffled" with the key scheduling algorithm (KSA), the key stream is generated by using and modifying this S-Array in the pseudo- random generation algorithm (PRGA). Cipher: In cryptography, for performing encryption or decryption a series of defined steps that can be followed as a procedure is known as Cipher algorithm. The types of cipher are as described below:

Fig 1 Generic Key-IV Model of a Stream Cipher

A. Block Cipher

In block cipher for encryption of data a cryptographic key and algorithm are applied to a block of data for example, 64 contiguous bits of block at once as a group rather than to one bit at a time. The representative diagram of block cipher structure is as shown in Fig. 1. It might be easier to decipher the cipher text because the blocks of text which are identical do not get encrypted the same way in a message, in a sequence it is common to apply the cipher text from the trailing encrypted block to the next block. So that on the same day identical cipher text does not produced by identical encrypted messages. From a random number generator an initialization vector is derived and combined with the text in the first block and the key. This ensures that all subsequent blocks result in cipher text that doesn't match that of the first encrypting.

B. Stream Cipher

In stream cipher a plaintext of one byte is encrypted at a time, although a stream cipher may be designed to operate on one bit at a time or on units larger than a byte at a time. The representative diagram of a stream cipher structure is as shown in Fig. 2. In this structure a key is input to a pseudorandom bit generator that produces a stream of 8-bit that are apparently random pseudorandom stream. An algorithm generates 8-bit stream which is unpredictable without knowledge of the input key. A key stream is called as the output of the generator, which this output of generator is combined one byte at a time with the plaintext stream using the bitwise exclusive-OR (XOR) operation. The encryption sequence should have a large period. A pseudorandom number generator uses a function which produces a deterministic stream of bits that eventually repeats. It will be more difficult to do cryptanalysis if the period of repeat is longer. The key stream should approximate the properties of a true random number stream as close as possible. For example, there should be an approximately equal number of 1s and 0s. All of the 256 possible byte values should appear approximately equally often if the key stream is treated as a stream of bytes. The advantage of a block cipher is that keys can be reused. However if two plaintexts are encrypted with the same key using a stream cipher then the cryptanalysis is
often quite simple. If the two cipher text streams are XORed together then the result is the XOR of the original plaintexts. The cryptanalysis may be successful if the plaintexts are text strings, credit card numbers, or other byte streams with known properties. If generator is properly designed then stream cipher can be as secure as block cipher of comparable key length.

The main advantage of a stream cipher is that stream ciphers are almost always faster and use far less code than do block ciphers. RC4 can be implemented in just a few lines of code.

III. RC4 ALGORITHM IMPLEMENTATION

The principle of RC4 algorithm mainly consists of two components: key-scheduling algorithm (KSA) and pseudo-random number generation algorithm (PRGA). The key function of KSA is the complete initialization of RC4 key while to produce pseudo-random number is the key function of PRGA. RC4 is a table based binary additive stream cipher. The output word of the key stream generator is used by the RC4 for its key stream. The word length n=8 is used for most of the applications. For a key stream generator RC4 is a unique design. Because of large internal memory of RC4 and the dynamic updating of tables the RC4 is secure from conventional attacks on key stream generators.

A). The Key Scheduling Algorithm (KSA) The key-scheduling algorithm initializes the permutation in the "S" array. The number of bytes in the key defines the "Key length" and can be in the range $1 \leq$ key length ≤ 256, typically between 5 and 16 bytes, corresponding to a key length of 40 – 128 bits. First, the array "S" is initialized to the identity permutation. For 256 iterations S is then processed in a similar way to the main PRGA, but at the same time also mixes in bytes of the key.

```
for i from 0 to 255
S[i] := i end for
j := 0
for i from 0 to 255
j := (j + S[i] + key[i mod key length]) mod 256
swap(&S[i],&S[j]) end for.
```

B.) The Pseudo Random Generation Algorithm: When large quantities of random digits are needed then a pseudo-random number generator (PRNG) program is written and used in, probability and statistics applications. The endless strings of single-digit numbers are produced by most of these programs usually in base 10, known as the decimal system. When large samples of pseudo-random numbers are taken, each of the 10 digits in the set {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} occurs hm (PRGA) with equal frequency, even though they are not evenly distributed in the sequence. In an attempt to produce truly random sequences of numbers many algorithms have been developed. Endless strings of digits is one in which it is theoretically impossible to predict the next digit in the sequence. For such machine-generated strings of digits this has given rise to the term pseudo-random. For most they are equivalent to random-number sequences, but according to the rigorous definition they are not truly random. Typically a sequence of zero and one bits that may be combined into sub-sequences or blocks of random numbers are produced by the Random Number Generators (RNGs) used for cryptographic applications. There are two basic classes of RNG: deterministic and nondeterministic. An algorithm that produces a sequence of bits from an initial value called a seed is known as deterministic RNG. A RNG which produces output that is dependent on some unpredictable physical source that is outside human control is known as nondeterministic RNG. There are no nondeterministic random number generators available which are FIPS approved. Pseudo random number generator (PRNG) belongs to the class of deterministic random bit generator (DRBG). PRNG is an algorithm which generates a sequence of numbers that approximates the properties of random numbers. The sequence which is generated by PRNG algorithm is not truly random in nature but it is completely determined by a relatively small set of initial values, called the PRNG's state. Although by using hardware random number generators the sequences that are closer to truly random can be generated. In practice pseudorandom numbers are important for simulations and are central in the practice of cryptography. From an arbitrary starting state a PRNG can be started. It will always produce the same sequence thereafter when initialized with that state. By the size of the state the maximum length of the sequence before it begins to repeat can be determined. However with each bit of 'state' added the length of the maximum period potentially doubles. For many practical applications it is easy to build PRNGs with periods long enough. If a PRNG's internal state contains n-bits its period cannot be longer than 2n results. Without walking through the whole period the period length for some PRNGs can be calculated. For the periods of exactly $2n-1$ the Linear Feedback Shift Registers (LFSRs) are usually chosen. Linear congruential generators have periods that can be calculated by factoring. Usually after walking through a non repeating starting sequence the mixes have periods of about $2n/2$ on average. Mixes that are reversible have periods of about $2n-1$ on average, and the original internal state will always included in the period. Although after PRNGs reach the end of their period they will repeat their results. A repeated result does not mean that the end of the period has been reached. Most pseudorandom generator algorithms produce sequences which are uniformly distributed by any of several tests. The security of most cryptographic algorithms and protocols using PRNGs is based on the assumption that it is infeasible to distinguish use of a suitable PRNG from use of a truly random sequence. The stream ciphers, which work by exclusive ORing the plaintext of a message with the output of a PRNG, producing cipher text are the simplest examples of this dependency. The design of cryptographically adequate PRNGs is extremely difficult because PRNG must meet additional criteria. In the cryptographic suitability of a PRNG the size of its period is an important factor, but not the only one.

IV. STEPS OF RC4 ALGORITHM

 ISSN: 2321-8169
186 - 191

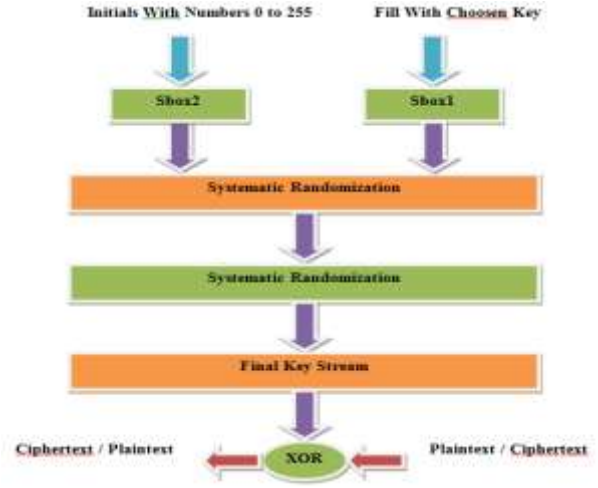

Fig. 2 Flow Diagram of Plaintext to Ciphertext

The steps for RC4 encryption algorithm are as follows:
1) Get the data to be encrypted and the selected key.
2) Create two string arrays.
3) Initiate one array with numbers from 0 to 255.
4) Fill the other array with the selected key.
5) Randomize the first array depending on the array of the key.
6) Randomize the first array within itself to generate the final key stream.
7) XOR the final key stream with the data to be encrypted to Give cipher text.

V. RESULTS

A. Synthesis Results of RC4 Algorithm of 256-bit Key size the synthesis result of RC4 algorithm is as shown below in table.1 for Virtex-5 FPGA kit. The system tested using confirmed test vectors in order to examine its correctness. The whole design was synthesized, placed and routed by using Virtex-5 XILINX FPGA device.

Place and Route Report:

No. of BUFGs1 out of 32	3%
No. of External IOBs 19 out of 640	2%
No. of LOCed IOBs0 out of 19	0%
No. of Slices3561 out of 17280	20%

Table. 1 Device Utilization Summary

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	2,093	69,120	3%	
Number used as Flip Flops	2,093			
Number of Slice LUTs	11,311	69,120	16%	
Number used as logic	11,303	69,120	16%	
Number using O6 output only	11,303			
Number used as Memory	8	17,920	1%	
Number used as Dual Port RAM	4			
Number using O6 output only	4			
Number used as Single Port RAM	4			
Number using O5 and O6	4			
Number of occupied Slices	3,840	17,280	22%	
Number of LUT Flip Flop pairs used	11,310			
Number with an unused Flip Flop	9,220	11,313	81%	
Number with an unused LUT	2	11,313	1%	
Number of fully used LUT-FF pairs	2,091	11,313	18%	
Number of unique control sets	6			
Number of slice register sites lost to control set restrictions	7	69,120	1%	
Number of bonded IOBs	19	640	2%	
Number of BUFG/BUFGCTRLs	1	32	3%	

No. of Slice Registers 2093 out of 691203%
No. used as Flip Flops 2093
No. used as Latches0
No. of Slice LUTS11310 out of 6912016%
No. of Slice LUT Flip-Flop pairs 11310 out of 69120 16%
Peak Memory Usage: 488 MB
Map Report:
No. of Slice Registers: 2,093 out of 69,1203%
No. used as Flip Flops: 2,093
No. of Slice LUTs:11,310 out of 69,12016%
No. used as logic:11,302 out of 69,12016%
No. using O6 output only: 11,302
No. used as Memory: 8 out of 17,9201%
No. used as Dual Port RAM:4
No. using O6 output only:4
No. used as Single Port RAM:4
No. using O5 and O6:4
No. of occupied Slices: 3,561 out of 17,28020%
No. of LUT Flip Flop pairs used: 11,310
No. with an unused Flip Flop:9,217 out of 11,310 81%
No. with an unused LUT: 0 out of 11,3100%
No. of fully used LUT-FF pairs: 2,093 out of 11,31018%
No. of unique control sets: 6
No. of slice register sites lost to control set restrictions:
7 out of 69, 1201%

B. Simulation Result of RC4 Algorithm for 256-bit Key size
The simulation result of RC4 algorithm is as shown in Fig 5. The simulation result as in Fig.5 shows that no security keys b generate for data encryption till 803 clock cycle. Because 803 clock cycle is required for key expansion i.e. for processing of key generation. For key scheduling algorithm 768 clock cycles are required and for pseudorandom number generation algorithm 3×n clock cycles are required where n is the number of bytes of plaintext. Here plaintext is of 1 byte (8-bit), hence 3 clock cycle is required for pseudorandom number generation algorithm. Here used key size is 32 byte (256-bit), therefore for loading of keys 32 clock cycle is required. Therefore total 768 + 3 + 32 = 803 clock cycles are needed for security key generation. Hence it can be seen that in above simulation result that after 803 clock cycle, security keys are generated but these keys are not used for encryption because these are weak keys. Weak keys are generated till the 1535 clock cycle. Hence till the 803 + 1535 = 2338 clock cycle output ready signal is deactive i.e. zero. This simulation window in Fig.6 shows that after 2338 clock cycle i.e. at 2339 clock cycle the output ready becomes high and security keys are available for data encryption. After 2339 clock cycles 8-bit keys i.e. 8c, 3c, 13, f8, c2.....are used for encryption of 8-bit data. For RC4 algorithm 256-bit keys are used for generation of 8-bit security keys. The 256-bit keys which are used here for m generation of security keys for data encryption is given below:
1a,da,31,d5,cf,68,82,21,c1,09,16,39,08,eb,e5,1d,eb,b4,62,27,c6,cc,8b,37,64,19,10,83,32,22,77,2a i.e. 32×8 = 256-bit.

 ISSN: 2321-8169
186 - 191

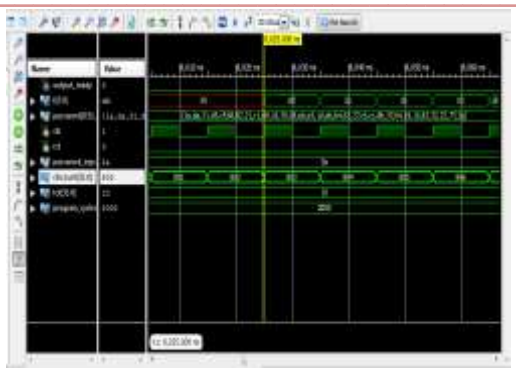

Fig 3(a) Simulation Result of RC4 Key Generation

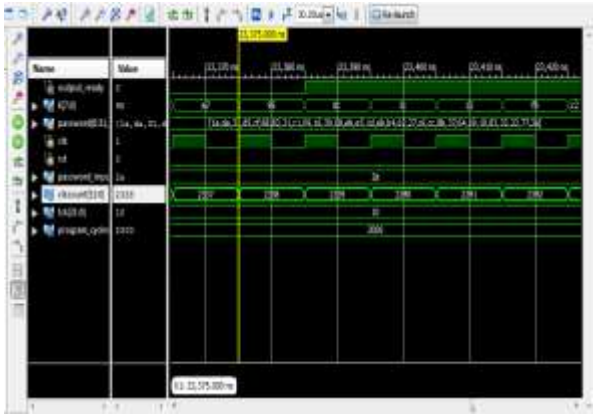

Fig. 4. (b) Simulation Result of RC4 Key Generation

C. Simulation Result of RC4 Algorithm for 56-bit Key size the synthesis report of RC4 algorithm is as shown in Table.2 when key size is reduced to 56-bit. The comparison of synthesis report for both key sizes is as shown in Table 3. After comparison of the synthesis report of key size 56 bit with the synthesis report of key size 256 bit, it is found that approximately same number of hardware devices is required for 56 bit key size. It means less key size does not affect hardware complexity but it reduces the security level. Because the probability to identify the permutation of 56-bit key size is high as compared to 256-bit key size. Therefore for high security level key size should be large. The simulation result of 56 bit key size RC4 algorithm is as shown in Fig.7. The simulation result as in Fig.7 shows that no security key generates for data encryption till 778 clock cycle. Because 778 clock cycle is required for key expansion i.e. for processing of key generation. It has been already discussed that for key scheduling algorithm 768 clock cycles are required and for pseudorandom number generation algorithm 3×n clock cycles are required where n is the number of bytes of plaintext. Here plaintext is of 1 byte (8-bit), hence 3 clock cycle is required for pseudorandom number generation algorithm. Here the key size is 7 byte (56-bit), therefore for loading of keys 7 clock cycle is required. Therefore total 768 + 3 + 7 = 778 clock cycles are needed for security key generation. Hence it can be seen that in above simulation result that after 778 clock cycle, security keys are generated while in 256 bit key size the security keys are generated after 803 clock cycle. It means that for large key size time required for generation of security Key stream also increases.

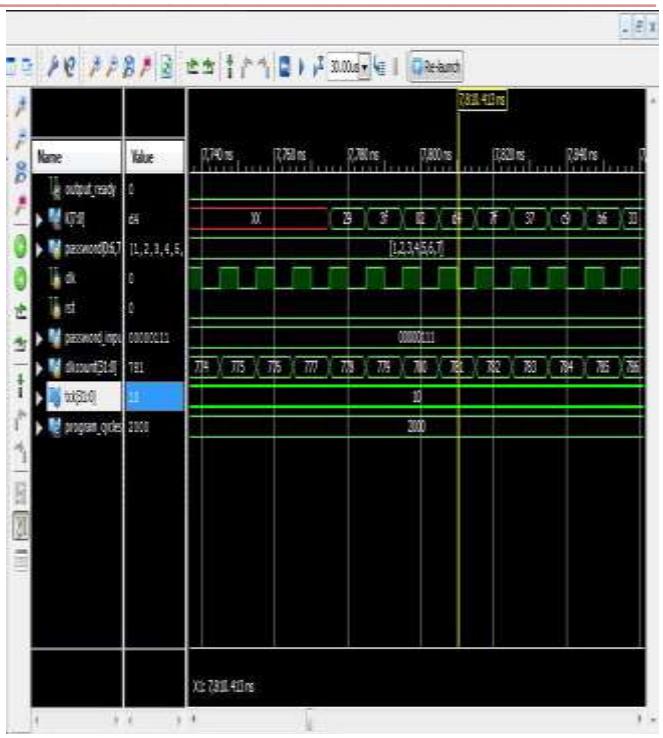

Fig. 5 Simulation Result of RC4 Key Generation (56-bit Key size

Table 3. Comparison of Synthesis Reports

Sr.No.	Name of devices	No. of Used Devices for Key size 56-bit	No. of Used Devices for Key size 256-bit
1	Slice Registers	2093	2093
2	Flip-Flops	2093	2093
3	Slice LUTs	11310	11311
4	O6 Output	11302	11303
6	LUT Flip-Flop Pairs	11310	11313
7	Fully Used LUT Flip-Flop Pairs	2093	2091
8	Bonded IOBs	19	19

Hardware complexity and key stream generation time point of view less key size is advantageous but security level point of view large key size is necessary. Because it is difficult for hackers to hack large key size as compared to less key size. The initial 1536 generated keys are discarded by keeping

output ready signal zero at the time of generation of these keys. After 1536 keys the next keys are available for ciphering, because at 2339 clock cycle output ready signal becomes high. Initial 1536 keys are discarded because these are weak keys. In the RC4 algorithm S register is filled by sequential number from 0 to 255. In key scheduling algorithm these numbers are shuffled 256 times. Key scheduling algorithm takes 768 clock cycles for its operation. As initially numbers are in sequential manner so it's initial shuffling does not producing much more complexity and hence hacker may hack the initial security keys. Therefore it is suggested that discard the initial 768 keys or it's multiple. So here multiple of 768 i.e. 1536 initial keys are discarded for increasing the security level. The speed of data encryption depends on the frequency of the system. As key scheduling algorithm needs 768 clock cycle, pseudorandom generation algorithm required 3 clock cycle and 32 byte key size loading required 32 clock cycle. Hence total 802 clock cycle required for key generation. After 802 clock cycle each one key generate per clock cycle. As first 1536 keys are discarded hence after 2339 clock cycle valid keys are available. The maximum frequency of the system is 135.247 MHz Here one clock cycle is of 10ns hence after 23.39µs the valid keys will be available for ciphering. Ones the valid key generation starts then data will be encrypted with the speed of 100 Mbps.

VI. CONCLUSION

A synthesis and simulation results of the RC4 stream cipher for wireless LAN Security is presented in this paper. The result shows that this design provides high data throughput using key length of 256 bytes. It provides high flexibility as it can be used in many applications with any key length from 1 byte to 16 bytes. It is observed that after 23.39µs the valid keys are available for data encryption. This system achieves a data throughput up to 100 MBytes/sec in a clock frequency of 135.247 MHz this paper describes the detailed simulation and synthesis results of RC4 algorithm which can be further implemented in FPGA based hardware. The effect of variation of key size is also analyzed and it is found that less key size does not affect the number of required hardware components so much. It can be seen that less key size required less time to generate key stream but security level decreases due to less key size. For 56-bit key size valid keys are generated after 23.14µs while for 256-bit key size valid key generation time is 23.39µs. RC4 is simple and cost effective algorithm and possible to implement in both hardware and software.

REFERENCES

[1] Rourab Paul, Amlan Chakrabarti and Ranjan Ghosh, "Hardware implementation of four byte per clock RC4 algorithm," in Journal of latex class files Vol. 6 No. 1, Jan. 2007.
[2] Jaya Dofe and Manish Patil, "Hardware implementation of modified RC4 stream cipher using FPGA," IOSRJEN, vol. 02, Issue 06, pp. 1447–1450, Jun. 2012.
[3] Poonam Jindal and Bramhajit Singh, "A survey on RC4 stream cipher," IJCNIS, vol. 7, pp. 37–45, Jun. 2015.
[4] Rajendar Racherla and S. Nagakishor Bhavanam, "Design and simulation of enhancing RC4 stream cipher for Wi-Fi securityusing Verilog HDL," IJERA, vol. 1, Issue 3, pp. 653–659.
[5] Sultan Weatherspoon, "Overview of IEEE 802.11b security," Network Communication Group, Intel Technology Journal Q2, 2000.
[6] IEEE Std 802.11. IEEE Standard: Hardware implementation of the RC4 stream cipher-P.kitsos, G. Kostopoulos, N. Sklavos and O.Koufopavlou.VLSI design laboratory.
[7] Claude E. Shannon. Communication theory of secrecy systems. Bell Systems Technical Journal, 28(4):656–715, 1949.
[8] BluetoothTM. Bluetooth specification, v4.0, June 2010. E0 encryption algorithm described in volume 2, pages 1072–1081. Available online at http://www.bluetooth.org.
[9] Yi Lu, Willi Meier, and Serge Vaudenay. The conditional correlation attack: A practical attack on Bluetooth encryption. In Victor Shoup, editor, CRYPTO, volume 3621 of Lecture Notes in Computer Science, pages 97–117. Springer, 2005.
[10] Yi Lu and Serge Vaudenay. Cryptanalysis of Bluetooth keystream generator two-level E0. In Pil Joong Lee, editor, ASIACRYPT, volume 3329 of Lecture Notes in Computer Science, pages 483–499. Springer, 2004.
[11] Yi Lu and Serge Vaudenay. Faster correlation attack on Bluetooth keystream generator E0. In Matthew K. Franklin, editor, CRYPTO, volume 3152 of Lecture Notes in Computer Science, pages 407–425. Springer, 2004.
[12] Yi Lu and Serge Vaudenay. Cryptanalysis of an E0-like combiner with memory. J. Cryptology, 21(3):430–457, 2008.