2nd International Conference on Emerging Trends in Engineering, Technology, Science and Management
Lakshmi Narain College of Technology, Bhopal
20th May 2017, www.conferenceworld.in
(ICETETSM-17)
ISBN: 978-93-86171-45-0

RC4 STREAM CIPHER DESIGN FOR DATA SECURITY

Vaishali Singh[1], Shriddha Shrivastawa[2]

[1]Student , [2]Assistant professor

Department of Electronics &Telecommunication Engineering,

Lakshmi Narain College of Technology, Bhopal, (India)

ABSTRACT

In this paper, an efficient algorithm of the RC4 stream-cipher is proposed to enhance the security when the network communication is established between software and hardware to transmit the information or data. One of the significant piece of this project is to generate simulation and synthesis report for RC4 encryption algorithm. The synthesis and simulation report is accomplished for Virtex-5 FPGA. Since, 0 to 256 bits can be used in Key Scheduling Algorithm (KSA) as the key size, so the key size of 256 bits is chosen for this project. The simulation result is verified using draft-josefsson-rc4-test-vectors-02 (This document consists of test vectors for stream cipher RC4).

***Keywords*: RC4, Stream Cipher, DES, KSA, PRGA, FPGA.**

I. INTRODUCTION

Cryptography is a Greek word for "hidden writing" The art and science of transforming (encrypting) information (plaintext) into a transitional form (cipher text) that stores information in storage or transit. The main feature of cryptography is to solve the problems, which are associated with verification, integrity and privacy. The Growing Use Of Networks Has Made Us Concern For The Security.The need of mechanism which guarantees the protection & privacy sends over the electronic media developed the term cryptography. A protocol is the sequence of actions, which is designed with two or more sides, through which a goal can be fulfilled. Cryptography also, is associated with the meaning of protocol. Thus, a cryptographic protocol is a protocol that deals with the use of cryptography. This protocol uses cryptographic algorithm and intends to halt attempts of thefts and invasions. By the use of Theory of Statistics and Theory of Numbers now a day's cryptography has been highly developed and strongly validated in science. Many kinds of cryptographic algorithms have been invented in order to solve the problems of cryptography. The complexity of these problems provides several categories of cryptographic algorithms. A much known cryptographic algorithm is the RC4 stream cipher. Normally, security occurs as a result of having a huge number of different transformations.. Cryptography, which is a part of cryptology, is further classified as secret codes versus ciphers. Unlike Steganography, which seeks to hide the existence of a message, cryptography seeks to deliver a message unintelligible even if the message is completely exposed. In 1987, Ron Rivest of RSA security designed rc4, consistently entitled as "Rivest Cipher 4", the RC acronym stands for "RON'S CODE". RC4 was initially a trade secret; still it was anonymously posted to the cipher punk's mailing list in September 1994, so a description of it

ISBN: 978-93-86171-45-0

leaked out in 1994. The aim of working of cryptography is to transmit the plaintext into a cipher text. Where the original data or message is called plaintext and encrypted data is called cipher text.

Plaintext + security key = cipher text.

Cryptology – Inherently a social science refers to the art of bridging this bizarre gap between an entirely social notion called 'security', and the logical foundations of computer science, mathematics and allied 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 conventional stream cipher. Moreover it is used by really important and famous protocols and standards such as WEP, WPA, TSL, SSL, etc. Another reason for this choice is that it is well known for its simplicity and efficiency.

The main objectives of this project are as follows:-

- To generate 8-bit security key stream to encrypt and decrypt the data.
- To Implement RC4 algorithm in Verilog language.
- To generate simulation and synthesis result for RC4 Stream Cipher Algorithm.

II. THEORY

This stream cipher was invented in 1987 by one of the originator of the RSA public key cryptography algorithm and establisher of RSA security i.e. Ron Rivest. Even though the RC4 CIPHER is officially named "Rivest cipher 4", it is also termed as "Ron's Code - 4". "RC - 2", "RC - 5" & RC - 6 are other encryption algorithm that also exists. The trade secret behind (supporting) RC4 was disclosed in September 1994 when the description of the cipher was sent to the Cypherpunks mailing list (lots of human interested in privacy and cryptography who used this mailing list to communicate). After that, the description was posted on many website and the genuineness of the information was confirmed as the resulting outputs of the described cipher were analogous the outputs of licensed RC4. RC4 had a really large success thanks to its simplicity and efficiency. It was used in many (countless) popular standards and protocols such as WEP, WPA, and SSL OR TLS.

2.1 STREAM CIPHERS

The domain of symmetric key cryptology contains two major cryptographic primitives – Block ciphers and Stream ciphers. In block cipher primitive encryption is carried out block-by-block by applying a key dependent transformation on a block of message bits at a time. Whereas, a key stream is a pseudorandom sequence of bits produced by a stream cipher, and encryption is done by masking or hide the plaintext (considered as a sequence of bits) by the stream cipher key stream; using a simple XOR operation in general. The cipher text thus obtained is also a sequence of bits of the same length as that of the plaintext. To increase the efficiency in software applications, modern day stream ciphers are adopting a word-oriented approach, where word-wise (size of the word depends on machine architecture) operations are used instead of bitwise operations. However, the fundamental philosophy remains the same.

2.1.1 ONE-TIME PAD AND PERFECT SECRECY

One-time pad (OTP), also called the Vernam cipher is the motivation behind stream ciphers. OTP is a truly random bit-string of the same length as that of the plaintext – unique for each plaintext – which is bitwise XOR-

(ICETETSM-17)
ISBN: 978-93-86171-45-0

ED with the plaintext to produce the cipher text. It is one of the strong technique of encryption that cannot be easily cracked. Shannon showed that OTP is perfectly secure, i.e. even for an adversary with unbounded computational power, it is impossible to derive any new information about the plaintext from the cipher text beyond what is possible via random guess. Intuitively, for a perfectly secret encryption, the cipher text reveals no information about the plaintext over prior knowledge. It is absolutely required that the masking (bitwise XOR) is done using a unique sequence of bits for each plaintext; owing to the name one-time-pad to achieve ideal information theoretic 'perfect secrecy' in OTP. As in OTP it is required to manage unique keys as large as the plaintexts therefore in reality the OTP scheme is not practical.

2.1.2. BLUETOOTH STREAM CIPHER

Bluetooth is one of the chiefly used modern technology for wireless communication, prevalent in an array of practical devices. In 1998, the Bluetooth special interest group (SIG) developed the technology. The technology has been embraced by all companies in the communication business ever since. The E0 stream cipher is used as a pseudorandom key stream generator for confidentiality in Bluetooth transmission. The cipher follows the standard design model of a combiner generator bythe use oflinear feedback shift registers, where the key length is typically of 128 bits.

- It comes up with the new shortrange radio link solution and provides strong protection in this range.
- The consumption of extra power is also limit by this technology.

2.1.3. GSM STREAM CIPHERS

A5/1 and a5/2 stream ciphers, designed around late 1980's which were used to provide privacy in the GSM cellular network. A5/2 is a debilitated version of A5/1, created for certain export regions. Both the ciphers A5/1 and A5/2, initially kept secret, became public in 1994 through leaks and reverse engineering. After several minor and major attackson A5/1 and A5/2 published during 1994–2006, the GSM association instructed that the GSM phones will not support A5/2 anymore, and usage of a5/1 was directed by the 3GPP association. Later in the 3G cellular systems, the key stream generation algorithm for privacy was modified to A5/3, which uses the block cipher KASUMI.

2.1.4. G STREAM CIPHERS

3GPP LTE advanced is the leading contender in the race towards 4G mobile technologies. For LTE advanced technology, the chosen security algorithms for encryption and authentication employ two different stream ciphers – snow 3G and ZUC. While in the earlier 3G platform snow 3G had already been deployed, along with KASUMI, the other cipher ZUC is a brand new design. Both the ciphers are based on similar design principles using word-oriented Linear Feedback Shift Registers; these are used in the LTE advanced technology within a portfolio, along with the block cipher AES-128

III. METHOD

The main features of cryptography is to solve problems like verification, privacy & secrecy.Many kinds of algorithm are invented to solve the problem of cryptography. Here we used RC4 stream cipher due to its attracting features and extensive usage.

Cryptography is further categorized in two categories, as follows

- Asymmetric

(ICETETSM-17)
ISBN: 978-93-86171-45-0

- Symmetric

○ Block Cipher

○ Stream Cipher

In an asymmetric cryptography, for the encryption of the information, it uses public key whereas at the decryption of the information, it uses private keys.

In a symmetric cryptography, for both the encryption and decryption process, a single key is used and that key should be known to both the parties.

Block cipher is a technique of symmetric cryptography which involves encryption of one block of text at a time. Where as in stream cipher, at a time, single bit encrypts.

Synchronous and Self-Synchronizing Stream Ciphers are the models which have been described so far, suits the class of synchronous stream ciphers. These stream ciphers, although most prominently studied and used in practice, have a practical drawback. Once the synchronization between the sender and the receiver is compromised, decryption of the cipher text does not yield the corresponding plaintext at the receiver end. In such a case, the ciphers at the sender and the receiver ends need to re-synchronize for subsequent operation. This requirement for re-synchronization poses practical problems while operating a synchronous stream cipher over a noisy channel, where (burst) noise may frequently destroy parts of the communicated cipher texts.There is another class of stream ciphers known as Self synchronizing stream ciphers, where the r^{th} key stream bit (word) depends on the key as well as on some predetermined number (t) of previous cipher text bits (words). So, the decryption works correctly if the previous (t) cipher text bits (words) are received properly, irrespective of earlier deletions.The selection of stream cipher depends on required level of security, speed of encryption and cost of system.

3.1 RC4 STREAM CIPHER

RC4 uses a variable length key from 1 byte to 256 bytes in order to initiate a 256-byte array. 256-bytes arrays include two types, S-Box and K-Box. The S-array is filled linearly such as S0=0, S1=1, S2=2……..S255 = 255. The K-array includes the key, repeating as necessary times, so that the array should get filled. The RC4 key is usually limited to 40 bits, for the reason of export restrictions but it is sometimes used as a 128 bits key. RC4 keys have the ability of using keys between 1 byte and 256 bytes. The RC4 algorithm works in two phases, the first phase is the key setup and the second phase is the pseudorandom key stream generation phase. The first and most difficult phase is Key setup in this algorithm. The encryption key is used to achieve an encrypting variable by utilizing twocategory of arrays, first S-array & second is K-array and N-number of mixing operations, during N-bit key setup (N being your key length). These mixing operations consist of swapping bytes according to RC4 algorithm. Figure 3 shows the flow diagram of the RC4 two phases. RC4 uses two counters, counter i and the counter j, these are initialized to zero value. In the key setup phase the S-box is being modified according to pseudo-code:

3.2 KEY SETUP PHASE

For i= 0 to 255

j = (j + Si + Ki) mod 256

ISBN: 978-93-86171-45-0

Swap Si and Sj

Once the encrypting variables are made from the key setup phase, then it enters into the pseudorandom key stream generation phase. The following pseudo code provides the pseudorandom key stream formation/generation phase as follows:

3.3 KEY STREAM GENERATION PHASE:

i = (i+ 1) mod 256

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 generate cipher text/plain text. XOR is the logical operation of comparing two binary bits. When the bits are different, the result is 1. When the bits are same, the result is 0. Once the receiver gets the encrypted message, he decrypts the encrypted message by XORing with the same encrypting variable.

3.4 KEY SCHEDULING ALGORITHM

The permutation array is generated with the help of the key-scheduling algorithm. The first step of this algorithm is to compute the *S* table with the identity permutation.The values which are in the array are equal to their index.Once the *S*array is initialized, the next step is to shuffle the array using the key to make it a permuted array

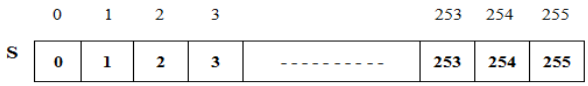

FIG.1. INITIALIZATION OF S TABLE

3.5 KSA ALGORITHM

For i = 0 to N – l

S[i] = i;

j = 0;

For i = 0 to N-1

{

j = (j + S[i] + k[i]) mod N;

Swap(S[i], S[j]);

i = i + 1

}

(ICETETSM-17)
ISBN: 978-93-86171-45-0

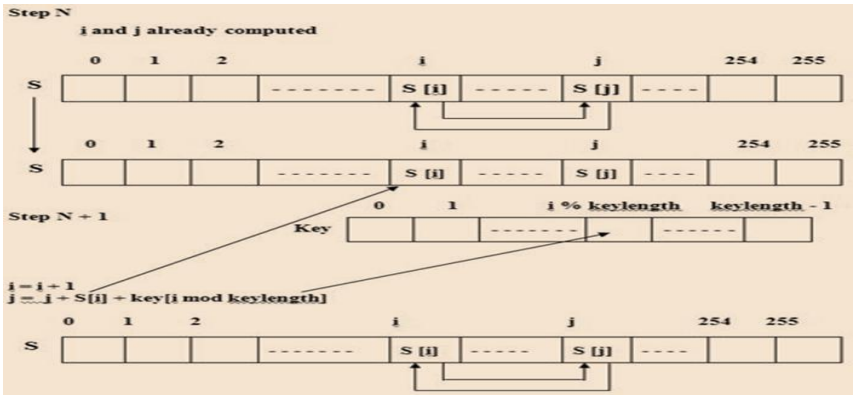

FIG. 2. SHUFFLING OF THE S ARRAY

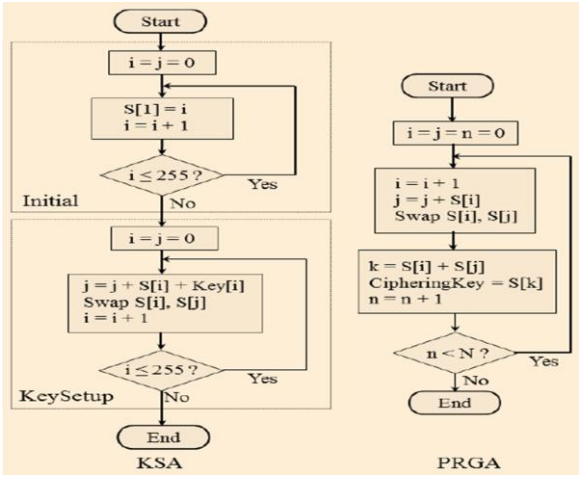

FIG 3. FLOW CHART OF RC4 ALGORITHM

3.6 PSEUDO-RANDOM GENERATION ALGORITHM

The algorithm consists of generating a key stream of the size of the message to encrypt. Firstly initialize the two indexes to 0 and then start the generation of the key stream one byte at a time until reached the size of the message to encrypt.

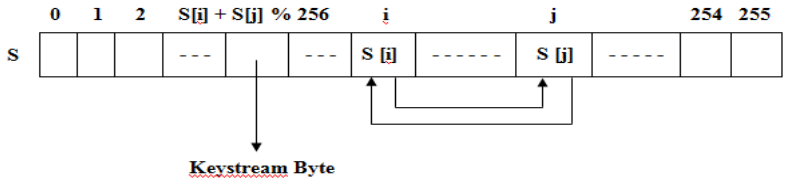

FIG.4. PR GENERATION ALGORITHM

i = j = 0;

Generation loop

{

i = (i + 1) mod N;

j = (j + S[i]) mod N;

Swap (S[i], S[j]);

ISBN: 978-93-86171-45-0

Output = S[(S[i] + S[j]) mod N];

}

IV. RESULT

To ensure Equations Security Key Generation

1. Synthesis Result.256 bit
2. Simulation Result. 256 bit.

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	3,846	17,280	22%	
Number of LUT Flip Flop pairs used	11,313			
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%	

FIG. 5. SYNTHESIS REPORT FOR 256 BIT KEY SIZE

(ICETETSM-17)
ISBN: 978-93-86171-45-0

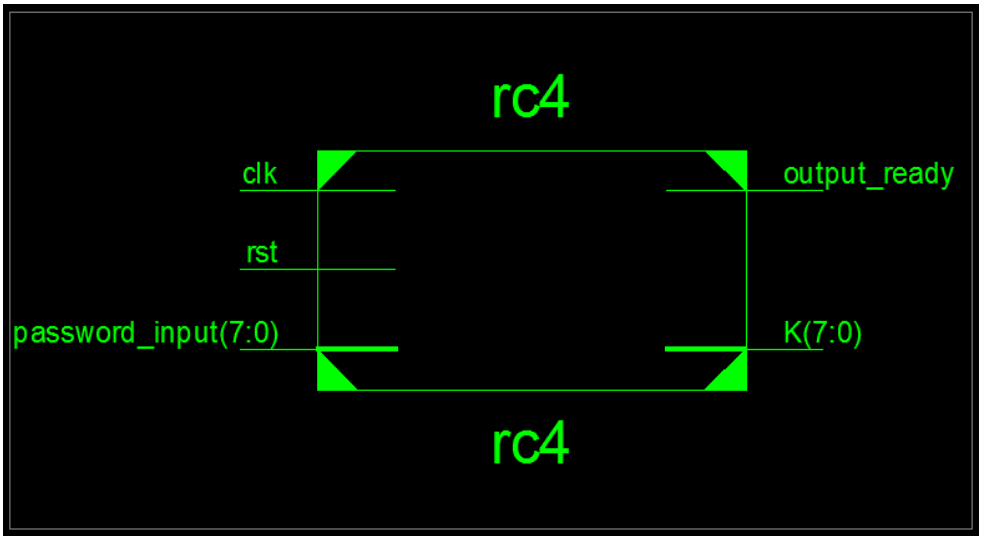

FIG.6. RTL VIEW OF "RC4" IMPLEMENTATION

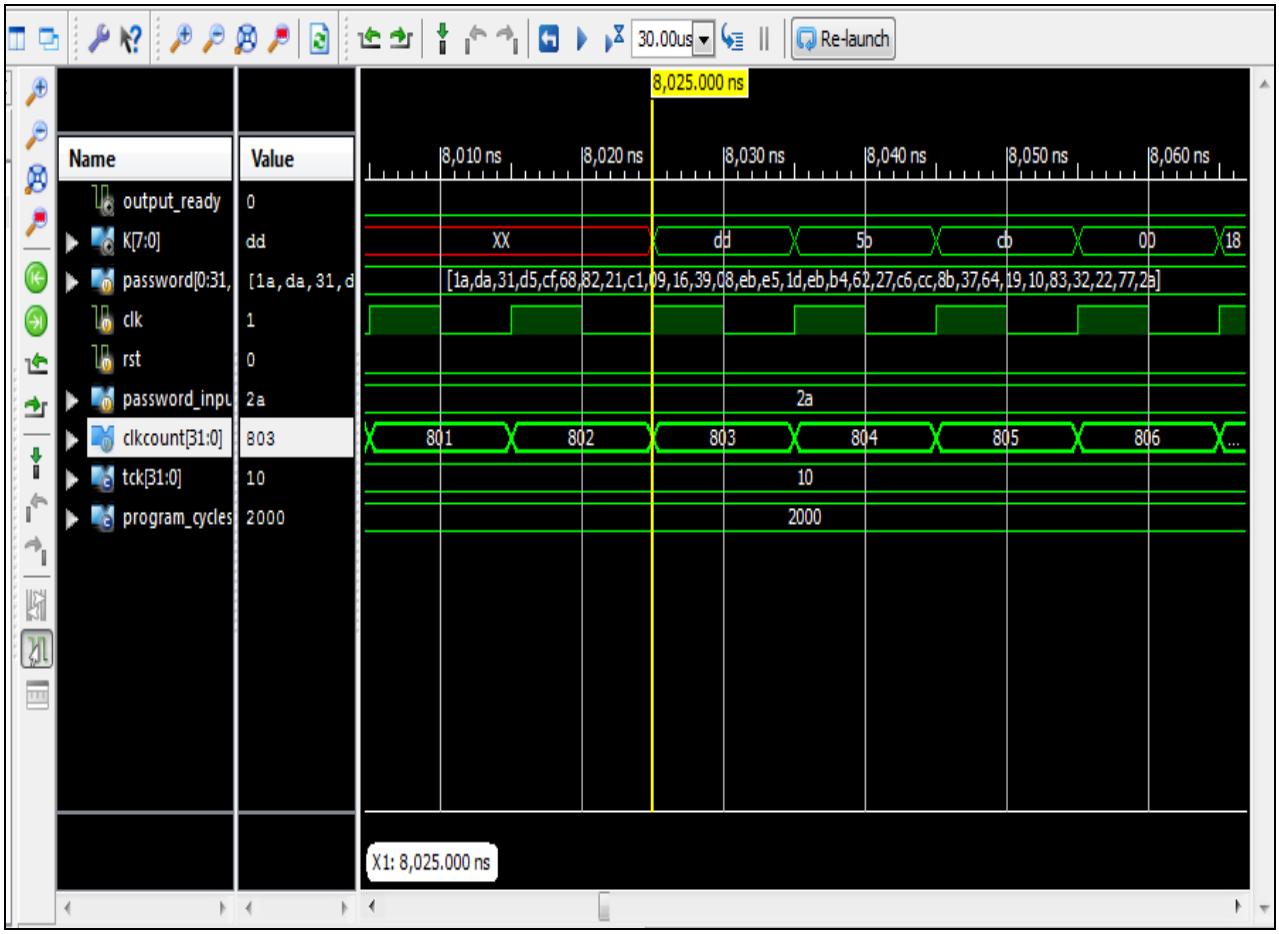

Fig.7. Simulation Result of 256 Bit Key Generations

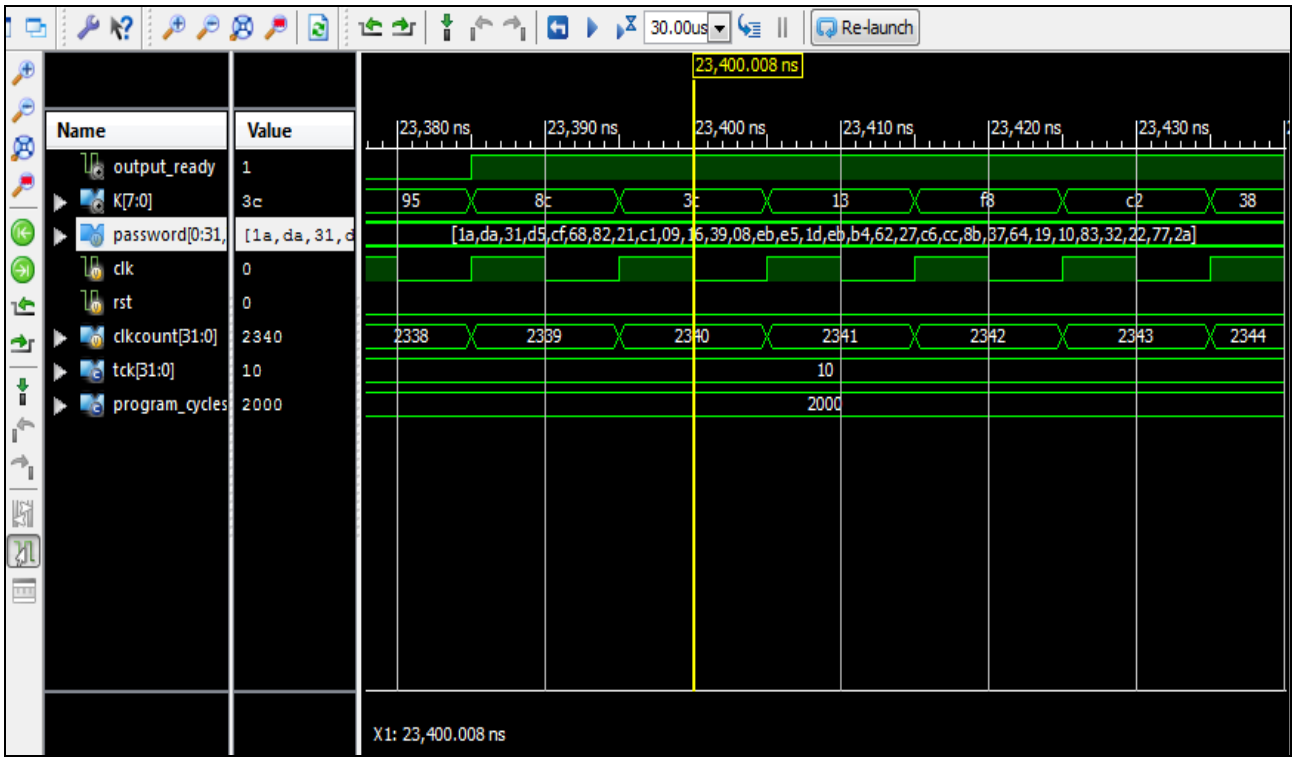

Fig.8. Simulation Result of 256 Bit Key Generations when Output Ready is One

The FPGA devices/element, enhances the security, reliability and performance. The encryption & decryption is performed stream wise and the data will remain secured in all respect. FPGAs are having the flexibility and high speed capability. They can be re-programmed to execute the more estimation intensive operations of a range of ciphers depending on security and application requirements. They also offer a better and cost effective solution than any ASIC or VLSI design of longer design cycle. Hardware implementation can be better than software in terms of high speed and level of security

V. CONCLUSION

This design provides high data throughput using variable key length from 0 bit to 256 bits. It provides high flexibility as it can be used in many applications. This system achieves a data throughput up to 100 Mbytes/sec in a clock frequency of 135.247 MHz often used in application where plaintext is present in quantities of unknown length. The developed algorithm consumes less bandwidth as we used the 8bit stream of data. This algorithm uses 1500 GHz and above frequency to work on which increases the level of security up to the great extent. Use of FPGA platform for the implementation of this project, the other platforms with various control elements can also be used. Like Embedded Systems, DSP Technology

REFERENCES

[1] *Hardware Implementation of RC4 Stream Cipher for Wi-Fi Security by* Vandana Malode , Nagnath Hulle, Of transaction 8654-34253-434-429 2014 IEEE

ISBN: 978-93-86171-45-0

[2] P. kitsos, G. Kostopoulos, N. Sklavos and O.Koufopavlou.VLSI design laboratory IEEE Std 802.11. IEEE Standard: Hardware implementation of the RC4 stream cipher.

[3] Claude E. Shannon. Communication theory of secrecy systems. Bell Systems Technical Journal, 28(4):656–715, 1949.

[4] Yi Lu, Willi Meier, and Serge Vaudenay. The conditional correlation attack: Apractical attack on Bluetooth encryption. In Victor Shoup, editor, CRYPTO, volumeYi Lu and Serge Vaudenay. Cryptanalysis of an E0-like combiner with memory. J.

[5] Yi Lu and Serge Vaudenay. Cryptanalysis of an E0-like combiner with memory. J.Cryptology, 21(3):430–457, 2008.

[6] Rourab Paul, AmlanChakrabarti and Ranjan Ghosh, "Hardware implementation of four byte per clock RC4 algorithm," in Journal of latex class files Vol. 6 No. 1, Jan. 2007.

[7] Jaya Dofe and Manish Patil, "Hardware implementation of modified RC4 stream cipher using FPGA," IOSRJEN, vol. 02, Issue 06, pp. 1447–1450, Jun. 2012.

[8] Poonam Jindal and Bramhajit Singh, "A survey on RC4 stream cipher," IJCNIS, vol. 7, pp. 37–45, Jun. 2015.

[9] RajendarRacherla and S. NagakishorBhavanam, "Design and simulation of enhancing RC4 stream cipher for Wi-Fi security using Verilog HDL," IJERA, vol. 1, Issue 3, pp. 653–659.

[10] Sultan Weatherspoon, "Overview of IEEE 802.11b security," Network Communication Group, Intel Technology Journal Q2, 2000.

[11] P. Hamalainen, M. Hannikainen, T. Hamalainen and J. Saarinen, "Hardware Implementation of the Improved WEP and RC4 Encryption Algorithms for Wireless Terminals", The European Signal Processing Conference (EUSIPCO'2000), September 5-8, 2000, Tampere, Finland, pp. 2289-2292.

[12] P. D. Kundarewich, S. J.E. Wilton, A. J. Hu, "A CPLD- Based RC-4 Cracking System", The 1999 Canadian Conference on Electrical and Computer Engineering, May 1999.

[13] K.H Tsoi, K.H Lee and P.H.W Leong, "A Massively Parallel RC4 Key Search Engine", Proc. of the 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'02), September 22 - 24, 2002 Napa, California, pp. 13-21.

[14] B. Schneier, D. Whiting, "Fast Software Encryption: Designing Encryption Algorithms for Optimal Software Speed on the Intel Pentium Processor", Fast Software Encryption workshop (FSE97), LNCS, Vol. 1267, pp. 242-259, Springer-Verlag, Haifa, Israel, January 20-22, 1997.

[15] "Recommendation for Block Cipher Modes of Operation. Methods and Techniques". National Institute of Standards and Technology, Technology Administration, U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/800-8a/sp800-38a.pdf

[16] Confirmed Test Vector for RC4, http://www.qrst.de/html/dsds/rc4.html

[17] Xilinx Inc., San Jose, Calif., "Virtex, 2.5 V Field Programmable Gate Arrays," 2003, www.xilinx.com