IF : 4.547 | IC Value 80.26

Volume-6, Issue-3, March - 2017 • ISSN No 2277 - 8160

Original Research Paper

Engineering

FPGA IMPLEMENTATION OF DYNAMIC KEY BASED STREAM CIPHER FOR CRYPTOGRAPHIC PROCESSOR

T.SRAVANTHI M TECH VLSI System Design, Dept of VLSI, ANURAG GROUP OF INSTITUTIONS, TS, India.

M.MURALI KRISHNA Assistant Professor, Dept of VLSI, ANURAG GROUP OF INSTITUTIONS, TS, India.

ABSTRACT The main objective of the project is to design dynamic key based stream cipher for hardware cryptographic applications such as information security and data transmission. Unlike a static key based existing stream ciphers, the novelty of this proposed stream cipher is based on dynamic key inorder to provide the data security. Public key is used as an input to the hash function which generates dynamic key using Toeplitz matrix and acts as an input to the RC4 algorithm. Further,this key is used to generate the dynamic hardware key for cryptographic processor. The Proposed method is Coded in Verilog, Synthesized and Simulated using Xilinx ISE Tool 13.2. We are designing our architecture in Verilog HDL code and also simulated by using Vivado 2015.2 tool and Hardware Implementation on ZYNQ Board(FPGA). The Project functionality input and output to the system is in digital data and the process expected to be done in the system is encryption and decryption. The specifications of dynamic key based stream cipher is of 8bits digital data input bit size and 8bits digital data output bit size.

KEYWORDS : hash function; key generation; hardware security; RC4 stream cipher; FPGA; HDL

I. INTRODUCTION

Information technology has become the backbone to the modern society, which makes data security an important aspect of research. Data need to be shared in such a format, which cannot be exploited in a feasible amount of time by any adversary without the knowledge of key parameters used to encrypt the information. Hence going for data encryption and decryption. Cryptography can be categorised by the type of encryption operations, the way in which plain text is processed with the number of keys used.

This paper confesses on the cryptographic system which exploits to generate the secured key of symmetric mechanism with the hash function. It provides security, performance and implements ability. In this same key is used for both encryption and decryption process. On the other hand, hash function is an algorithm that maps the data of arbitrary length to a fixed length data. Due to the one wayness of hash functions and less hardware complexity, this is preferred solution to many cryptographic applications where authentication of receiving data is the goal.

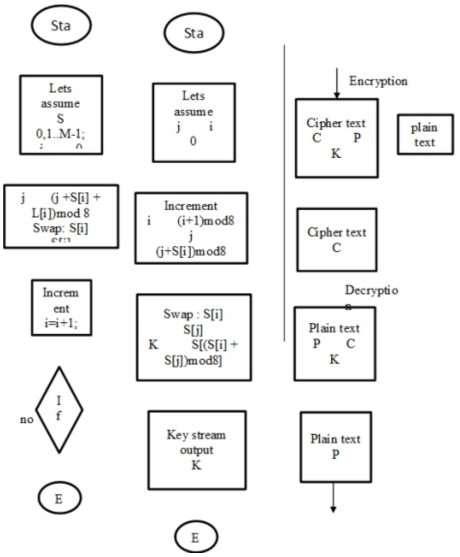
In the communication system using RC4 algorithm, the secret key is shared between the transmitter and receiver. In this paper, a method of secured encryption and decryption using RC4 algorithm and Toeplitz hash function in the integrated form is proposed where key is generated using a toeplitz matrix based hash value generator. This method is designed using Hardware Description Language (HDL) and implemented on FPGA device.

II. DESIGN APPROACH

1. RC4ALGORITHM

Rc4 stream cipher is proposed and designed by Ron Rivest for RSA security of data. The improved and effective version of the RC4 stream cipher is postulated by jian xie et al and weerasinghe. In cryptography, RC4 (also known as ARC4 or ARCFOUR meaning Alleged RC4) is the most widely-used software stream cipher and is used in popular protocols such as secure sockets layer (SSL) (to protect Internet traffic) and wired equivalent privacy (WEP) (to secure wireless networks). RC4 is remarkable for its simplicity and speed in software. RC4 generates a pseudorandom stream of bits (a key stream) which, for encryption, is combined with the plaintext using bit-wise XOR operation; decryption is performed the same way (since exclusive-OR is a symmetric operation).

The RC4 Stream cipher consists of substitution s-box ,S an array of length M, every position of s can store one byte of information. To scramble this permutation of all the possible 8 bytes ,secret key k of l-bytes size (typically,5≤ l ≤16) is required. An array l of length m clutches the core key ,with the secret key k repeated as L[i] = k[i mod l], for 0 ≤ i ≤ M -1.In a communication system, the secret key is shared between the receiver and the transmitter. at the transmitter ,encryption is done using simple XOR operation(C =P K) to get the cipher text (C) and the decryption at the receiver side is done using XOR operation of the cipher text (P = C K) to retrieve the original plain text text(P).



xt (C) and the decryption at the receiver side is done using XOR operation of the cipher text (P = C K) to retrieve the original plain text text(P).

Figure 1: Algorithm state machine to construct RC4 algorithm stream cipher: key scheduling phase, pseudo random generation phase, stream cipher.

The pseudo random generation (PRG) and the key scheduling (KS) phases are the main components of RC4 hardware design algorithm is represented in the algorithm state machine as shown in the figure 1. Key scheduling phase itself consists of two sub-stages, namely initialization and key setup .In the initialization stage, an M-element S-box is generated where M lies in the range from 0 to 2 power (n-1),where n is the size of the key. The initialized S-box is then permuted based on the provided key L in the key setup stage where

M number of iterations takes place. Then the PRG phase uses this S-box to generate pseudo random key stream bytes, where are of an arbitrary number. The steps and process of KS, using the Toeplitz hash function elucidated below.

2 .TOEPLITZ HASH FUNCTION

Hash algorithm is a type of cryptographic primitives. Hash algorithm takes as input a message of arbitrary length and produces a hash value or message digest as output. In this, hash values are generated by multiplying a message string with random binary matrix. Let E be a x * y matrix with Boolean values, where y and x are message size and hash length in bits respectively. Let y be a message consisting of y bits and Boolean multiplication of the matrix E by the column matrix of the message y as the hash value h(E).The matrix is uniquely determined for the first row and coloumn;the hash function H uses the toeplitz matrix requires x + y-1 bits determines the whole matrix.the hash function H family is - linear if for all arguments E,E' satistfy h(EE') = h(E) ^ h(E'),are feedback for the function in H family,where h€p H.the hash function is called €-balanced for €=2power (-x) if P(r)(h(E) =c) ≤ €, E ≠ 0,c Where p(r)(h(E)) represents probability of the event h(E).The characteristic property of the toeplitz matrix is that each value of the columns in the binary matrix is achieved by moving down the values of the preceding column and adding the new value element to the first element of the coloumn,thus in the toeplitz matrix each successive element of the column embodies the successive stages of the linear type of feedback shift register.FSR with the feedback polynomial of degree n, which is used to define the toeplitz matrix, with the initial seed value of n bits. The steps to construct the hash value can be well explained below:

The irreducible polynomial of degree n over GF (2) is represented as g(y) ,and LFSR generates the sequence of bit a0,a1....... with the feedback polynomial g(y) and the initial seed value is represented as a0,a1.....an.Thus for every irreducible polynomial g(y) and initial value of state a≠0,the hash function is associated with the message Y consisting of y bits of binary length as a linear function of the combination .Lets consider the irreducible feedback polynomial for generating the hash value using the LFSR polynomial g(y) = x8 +x4+x3+x2+1 and the initial seed value of LFSR is considered as (a0,a1....a7) = (0 1 0 1 0 0 1 0).

To determine the requirement of the hash function in the toeplitz matrix, should satisfy x + y -1 bit, the output sequence from LFSR should produce 8 bits where x is the degree of g(y) is 8 and the message bits y=8.with the irreducible polynomial g(y) of degree 8 and with initial seed value, the 15 bit sequence of LFSR output is (a0,a1........a14) = (0 1 0 1 0 0 1 0 1 0 0 0 1 0 0).

LFSR Output Sequence = [0 1 0 1 0 0 1 0 1 0 0 0 1 0 0]

Toeplitz matrix (8 x 8) =

↓ ↓ ↓ ↓ ↓ ↓ ↓

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

For each value of the column in the binary matrix is achieved by moving down the values of the preceding column and adding the new value element to the first element of the column ,thus in the toeplitz matrix each successive element of the coloumn embodies the successive stages of the linear type of feedback shift register is achieved after every clock cycle to top of the each column.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T$$

Lets us assume the message bits as 11001111,thus multiplying the message in column matrix with the toeplitz matrix which is obtained with the LFSR sequence ,the hash value output is 01101000.

The architecture of the hash value generation is given in the figure: 2.

The continuous stream of public key (seed value) is fed as an input to the hash function block to generate the stream of the hash value. The key is fed to generate the hash value by right shifting the operator bit by bit.

LFSR based PRNG output bits are multiplied with the input bits,which can be achieved by using AND gate and passed on to the MAC unit so as to accumulate it over a specified period of time. The output is pushed every nth clock cycle using a multiplexer which is controlled by a control unit.

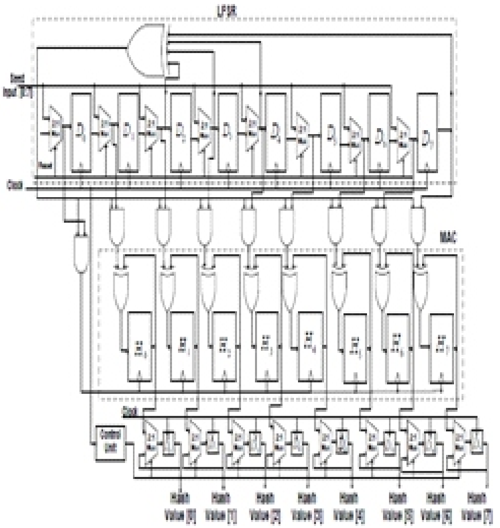


Figure:2 bit architecture and implementation of LFSR based toeplitz hash function.

LFSR based PRNG output bits are multiplied with the input bits, which can be achieved by using AND gate and passed on to the MAC unit so as to accumulate it over a specified period of time. The output is pushed every nth clock cycle using a multiplexer which is controlled by a control unit. The output of the hash function is fed as an hash value (i.e., dynamic key) to the stream cipher block.

III.PROPOSED METHODOLOGY

The proposed model consists of toeplitz hash function and RC4 algorithm is used in the integrated form. The public key is used as an input to the hash function which generates the hash value using toeplitz hash function.

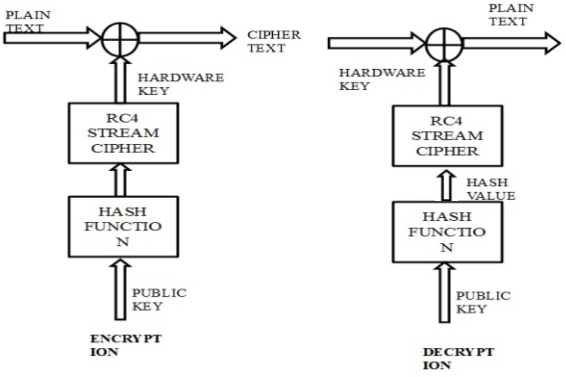


Figure:3 proposed model for encryption and decryption.

The hash value is applied as an input to the stream cipher which generates the hardware key. The hardware key is used to encrypt the plain text message using XOR operation.

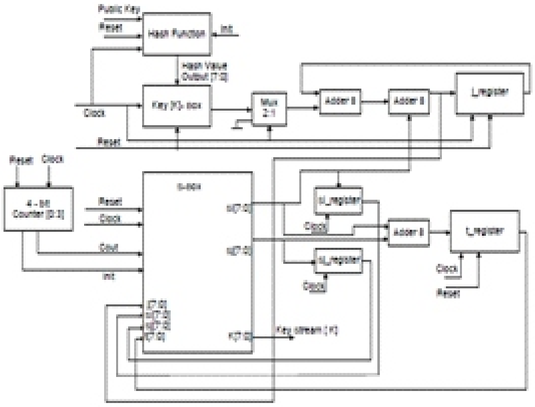


Figure:4 Schematic Of Proposed Design

In this proposed design ,dynamic key stream is constructed where the key k stream is replenished on every n cycle by a hash value of the public key as shown in figure:4.LFSR based toeplitz matrix is used to generate the hash value. In general, a fixed x * y toeplitz matrix is used to generate the hash value where the LFSR is reset after every computation of the hash value. But since the input to the hash value generator is the public key k in every cycle, this method would generate the same hash value every time.

III. TEST RESULTS

A. Simulation Results

I. Encryption

The encryption module is coded in Verilog and Simulated by using Vivado Tool.

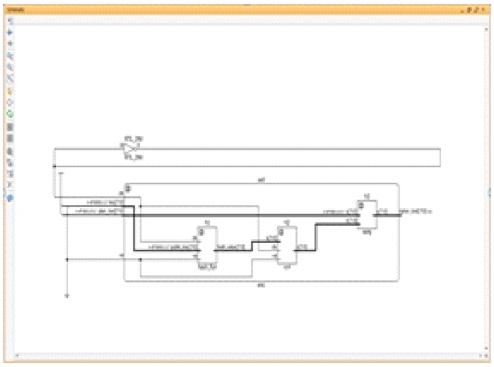


Figure 5: RTL Schematic view for Encryption

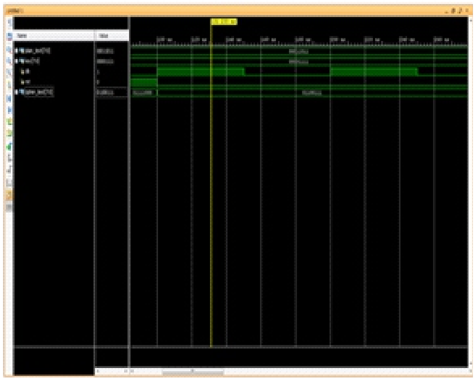


Figure 6 : simulation result for encryption.

S.no	Logic	Logic utilization in proposed method	
1	No. of slices	30	
		Slice LUT's	75
		Slice registers	24
2	No. of flip-flops	77	
3	No. of LUT's	LUT's as logic	75
		LUT's flipflop pair	83
4	No. of bonded IOB's	26	

Interpretation: From the above simulation,when the inputs are plain text = 00011011; key = 00001111; clock(clk) = 1; reset (rst)= 0 then the output of the encryption module is cipher text=01100111.

ii. Decryption

The decryption module is coded in Verilog and Simulated by using Vivado Tool AND implemented in Zynq Board (FPGA).

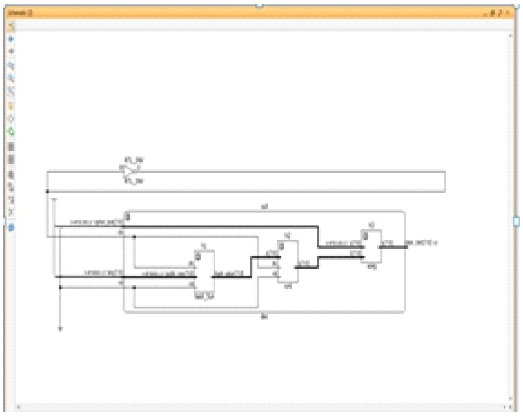


Figure 7: RTL Schematic of decryption Module

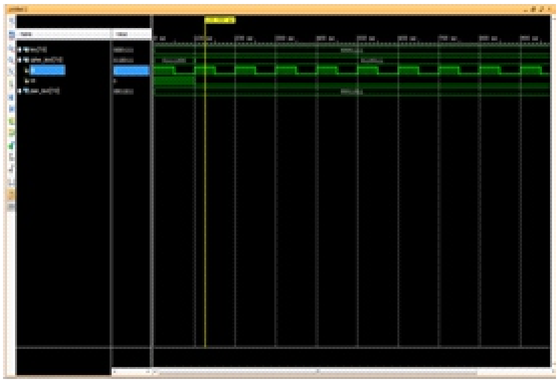


Figure:8 simulation result for decryption module.

Interpretation: From the above simulation waveform, when the input key =00001111;cipher text=01100111 then the output of the decryption Module is plain text=00011011. The representation of input and output is in the binary form as seen in figure 9.

IV. RESULT ANALYSIS

The following are Proposed design logic utilization table:

Table:1 logic utilization

The result shows that better performance in proposed design implementation in ZYNQ Board(FPGA) 7000 series was implemented in family XC7Z020 with package number CLG484 and speed grade is -1.Vendor is Xilinx.

Advantages

- Improvement in security.
- To acquire good efficienc

- Faster
- Simple to implement

Applications

- Used in network protocols like SSL,TLS.
- Rekeying mechanism to improve the security.· Communication systems.

V. CONCLUSION

The new design of dynamic key based stream cipher for cryptographic processor is designed ,simulated in Vivado tool and implemented on zynq board. unlike the static key ,in this dynamic key is used such that security is improved to the greater extent.

Future Scope

In future, we can look into the hardware architecture for optimizing the area without compromising the runtime performance. It is also an interesting research to look into a flexible hardware platform, which can support multiple stream ciphers.

REFERENCES

[1] N. Sklavos P. Kitsos, G. Kostopoulos and O. Koufopavlou., "Hardware implementation of the RC4 stream cipher," IEEE 46th Midwest Symposium on Circuits and Systems, vol. 3, pp. 1363–1366, Dec 2003.

[2] K. Zeng, C. -H. Yang, D. -Y. Wei, and T.R.N. Rao, "Pseudorandom bit generators in stream-cipher cryptography," IEEE Transactions on Computers, vol. 24, no. 2, pp. 8–17, Feb. 1991.

[3] S.S. Gupta, A. Chattopadhyay, K. Sinha, S. Maitra, and B.P Sinha, "High-Performance Hardware Implementation for RC4 Stream Cipher," IEEE Transactions on Computers, vol. 62, no. 4, pp. 730–743, 2013.

[4] Krawczyk H., "LFSR-based hashing and authentication," In: Advances in cryptology - crypto'94. Lecture notes in computer science, SpringerVerlag, vol. 839, pp. 129–139, 1994.

[5] P.P.Deepthi and P.S. Sathidevi, "Design, Implementation and analysis of hardware efficient stream ciphers using LFSR based hash functions," Computers and Security, vol. 28, no. 3-4, pp. 229–241, 2009.

[6] S. Pal, K.K. Soundra Pandian, and K.C. Ray, "FPGA implementation of stream cipher using Toeplitz Hash function," in International Conference on Advances in Computing, Communications and Informatics (ICACCI '14), Sept. 2014, pp. 1834–1838.

[7] Chang, D., Gupta, K., Nandi, M, "RC4-Hash: A New Hash Function based on RC4 (Extended Abstract)," In: Barua, R., Lange, T. (eds.) INDOCRYPT '06. LNCS, Springer, Heidelberg, vol. 4329, 2006.

[8] B. Zoltak, "VMPC one-way function and stream cipher," In B. Roy and W. Meier, editors, Fast Software Encryption,Lecture Notes in Computer Science,Springer-Verlag, vol. 3017, pp. 210–225, 2004.

[9] Maximov, Alexander, "Two Linear Distinguishing Attacks on VMPC and RC4A and Weakness of RC4 Family of Stream Ciphers," Fast Software Encryption, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 342–358, 2005.