Performance analysis of Enhanced Blowfish over Advanced Encryption Standard for Secure Integrated Circuits

2 authors:

V. Kumara Swamy
Sreenidhi Institute of Science and Technology
10 PUBLICATIONS **7** CITATIONS

SEE PROFILE

P G BENAKOP
Methodist College of Engineering and Technology
37 PUBLICATIONS **137** CITATIONS

SEE PROFILE

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 8, Issue 3, March 2018)

Performance analysis of Enhanced Blowfish over Advanced Encryption Standard for Secure Integrated Circuits

V. Kumara Swamy[1], **Prabhu Benakop**[2]
[1]*Associate Professor, Dept of ECE,Sreenidhi Institute of Science and Technology, Hyderabad, Telangana-501301*
[2]*Professor, Dept of ECE, Indur Institute of Engineering and Technology, Siddipet, Telangana, India-502103*

Abstract-Computer data communication networks handling important data and proving security on wireless and wired data communication channels. Security of the data transferred over the computer networks is more concerned to maintain secrecy in information, may be public or private relevant information. Hackers are trying hard to crack the software key and indulge in cyber crimes. In this paper, this paper main goal is to provide security to the data transferred at the software level as well as hardware level by proposing the enhanced Algorithm which can result in high speed, high throughput and effective memory utilization. Proposed Blowfish (BF) algorithm with architectural modifications called Enhanced Blowfish (EBF) improved its throughput compared to Blowfish (BF) and Advanced Encryption Algorithm (AES) algorithms. In the proposed three implementations, EBF algorithm produced minimum delay and throughput compared to BF and AES. It is implemented with bottom up approach using industry standard Mentor Graphics ModelSim, Leonardo Spectrum and Xilinx ISE Electronic Design Automation (EDA) tools. It is implemented using Verilog Hardware Description Language (HDL).

Keywords-BF, AES, EBF, VLSI, EDA, HDL

I. INTRODUCTION

Data encryption and decryption is challenging task for the design engineers to protect the data from hackers. Hackers are trying very hard to crack it so as to steal the information related to a design, an organization, a person or a bank account. Various cryptographic algorithms came in to existence, but BF and AES are the two best security algorithms used by information security agencies. The brief introduction to these to algorithms is given below:

A. Advanced Encryption Standard Algorithm (AES)

AES is a block cipher with variable key length. The block length and key length may be 128/192/256 bits with 9/11/13 rounds respectively. Each processing round consists of four steps, i.e., substitute bytes, shift rows, mix columns and add round key. AES encryption is flexible, more secured and fast [9, 18].

B. Blow-Fish (BF)

This algorithm is a symmetric algorithm with variable encryption key length. The plain text width is in 64-bit. The key length varies from 32-bits to 448-bits. Encryption is performed using 16-round fiestel network. Each round performs key dependent operations such as XOR, ADD AND SUBSTITUTE etc on plain text. It is faster than TDES and AES. It’s a replacement for DES algorithm [1].

II. RELATED RESEARCH REVIEW

RAM2016 presents field programmable gate array (FPGA) based blowfish algorithm to reduce power consumption and improve throughput. Analysis is aimed increase the speed, improve the throughput and reduce power consumption [1].

NMA2016 presents Blowfish and Rivest Cipher 6 (RC6) hybrid algorithm. The proposed algorithm improved the blowfish security and the efficiency. It eliminates the collision attack problem. Sub key generation process removes the Brute Force attack. It takes less encryption-decryption time and increases throughput [2].

VS2015 paper proposed RSA and blowfish encryption hybrid algorithm. It has symmetric and asymmetric processes. It is most preferred algorithm for cloud computing process. It is implemented using VHDL [3].

KYD2014 presents blowfish algorithm yielded minimum propagation delay and thus more throughput with the optimization technique used for reducing the encryption time. It is implemented on FPFA using VHDL [5]

ATM2014 proposes new approach in this paper which generate S-boxes and P-arrays with less time and provides same level of security compared with original blowfish algorithm [6].

III.THEORETICAL ANALYSIS OF CRYPTOGRAPHIC ALGORITHM

The most popular and secured cryptographic algorithms on them the software industry is relaying are AES and BF algorithms. The theoretical analysis of these two algorithms is described in the following sections.

A. Advanced Encryption Standard (AES)

AES is a symmetric key algorithm. The plain text and encryption key length may be 128/192/256 bits with 9/11/13 rounds respectively [7, 10, 19]. Round keys are used in every round of encryption process. Round keys are specially derived keys. Round keys are applied, along with other operations on block of data to be encrypted called as state array.

The following steps to be followed in encryption process of AES algorithm:

1. Round keys generation from the cipher key.
2. Initialization of the state array with plaintext.
3. Addition of initial round key to state array.
4. Nine rounds of state manipulation operations to be performed.
5. Tenth and final round of state manipulation to be performed.
6. Ciphertext is the final state array as the encrypted data output.

AES algorithm with 128-bits plain text performs operations on four rows and four columns of byte array or state array. This 128-bit plain text is numbered as D0 to D15 bytes represented in an array format. Each round of the encryption process follows four operations. They are Sub Bytes, Shift Rows, Mix columns and Xor Round key. All these operations are performed on the present state array and produce a updated state array. During nine rounds, the mentioned four operations are performed. In the tenth round, except Mix Columns operation, other three operations are done.

Sub Bytes

This operation is a simple substitution that converts every byte into a different value. AES defines a table of 256 values for the substitution. We need to work through the 16 bytes of the state array, use each byte as an index into the 256-byte substitution table, and replace the byte with the value from the substitution table. Because all possible 256 byte values are present in the table, it ends up with a totally new result in the state array, which can be restored to its original contents using an inverse substitution table. The contents of the substitution table are the entries computed using a mathematical formula but most implementations will simply have the substitution table stored in memory as part of the design [7].

Shift Rows

In this operation, each row of state array is rotated by number of bytes to the right row-wise from first to fourth i.e. rotated by 0 bytes, rotated by 1 byte, rotated by 2 bytes and rotated by 3 bytes respectively.

Mix Columns

The Mix Columns is explained with respect to the matrix multiplication shown in fig.1. It takes each column of the state array C0 to C3 and replaces it with a new column computed by the matrix multiplication [10].

$$\begin{bmatrix} C'_0 \\ C'_1 \\ C'_2 \\ C'_3 \end{bmatrix} = \begin{bmatrix} 02 & 01 & 01 & 03 \\ 03 & 02 & 01 & 01 \\ 01 & 03 & 02 & 01 \\ 01 & 01 & 03 & 02 \end{bmatrix} \bullet \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}$$

Figure.1: Mix Columns Operation

XOR Round Key

It takes the existing state array, XORs the value of the appropriate round key, and replaces the state array with the result. It is done once before the rounds start and then once per round, using each of the round keys in turn. The AES Encryption process is given in the below flowchart in fig.2.

Decryption Process

Decryption is reverse of encryption process. It performs XorRoundKey, InvShiftRows and InvSubBytes operations in the initial decryption round whereas it performs XorRoundKey, InvMixColumns, InvShiftRows and InvSubBytes in the nine full decryption rounds. XORing twice gives back to the same original value, so InvXorRoundKey is not required. It performs final XorRoundKey operation to produce the plain text. The same round keys used at encryption are used in the decryption as well in the same order. The fig.2 gives clear idea about the process of AES Encryption algorithm [7].

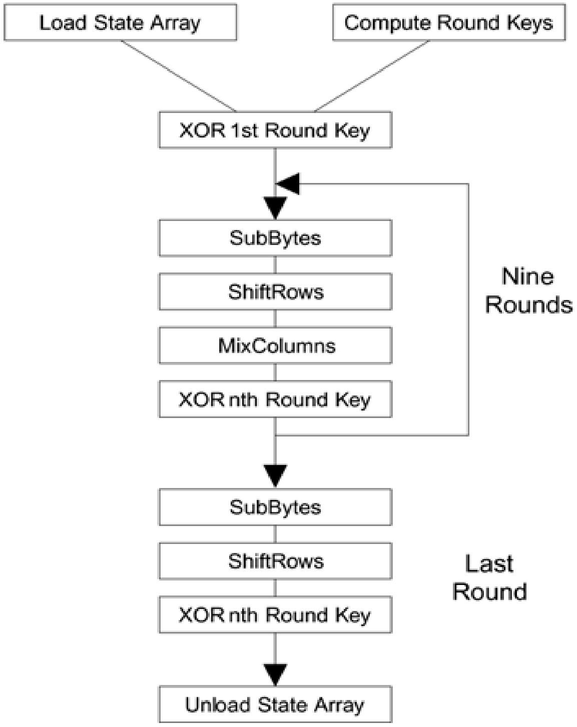

Figure.2: AES Encryption Process

B. Blow-Fish (BF)

Blowfish algorithm is a symmetric key block cipher. It is more secured than AES, DES and 3DES algorithms. It is fast, compact, simple and secure algorithm compared to other symmetric key algorithms. It has variable key length of 32 to 448 bits [1].

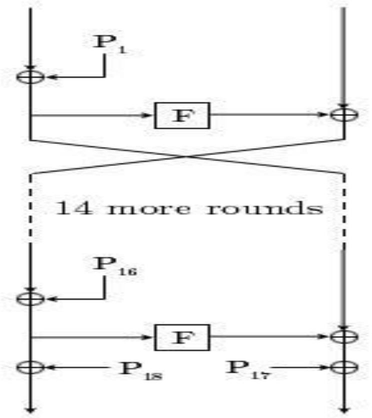

Figure.3: The Fiestel structure of Blowfish

Description of Algorithm

It is a symmetric key block cipher having 64-bit data as plain text and key of maximum 448-bit length. This algorithm is divided into two parts, i.e. Key-expansion and Data Encryption.

Key-expansion

It will convert a key of at most 448 bits into 14-sub keys (K1 to K14), each of them with 32-bits. These keys are generated separately and they form the p-array consists of 18, 32-bit sub keys i.e. P1, P2… P17 and P18. They are generated with the following logic:

P1=P1^K1,P2=P2^K2…P14= P14^K14,

P15=P15^K1,P16=P16^K2,P17= P17^K3, P18= P18^K4. ------------- (1)

The fiestel function consists of four 32-bit S-Boxes; each consists of 256 entries i.e. [0:31] S1 [0:255]
[0:31] S2 [0:255]
[0:31] S3 [0:255]
[0:31] S4 [0:255]

Data Encryption

It has 16- rounds of operations. Each round consists XOR, Fiestel function, XOR and swap operations as shown in fig.4. Operations involve XORs and additions on 32-bit Left Encryption (LE) and Right Encryption (RE) [4]. Fiestel function consists of four s-boxes each of 256 rows of 32-bits each. This algorithm is described in flowchart for in fig.3 and encryption process is given in fig.4. The process is in the form of an algorithm as given below:

```
Divide x into two 32-bit halves: LE, RE
    For i = 1 to 16:
        LE = LE XOR Pi                    RE = F(LE)
XOR RE
        Swap LE and RE                    Swap LE and RE
(Undo the last swap.)
        RE = RE XOR P17
        LE = LE XOR P18
Recombine LE and RE …… (2)
```

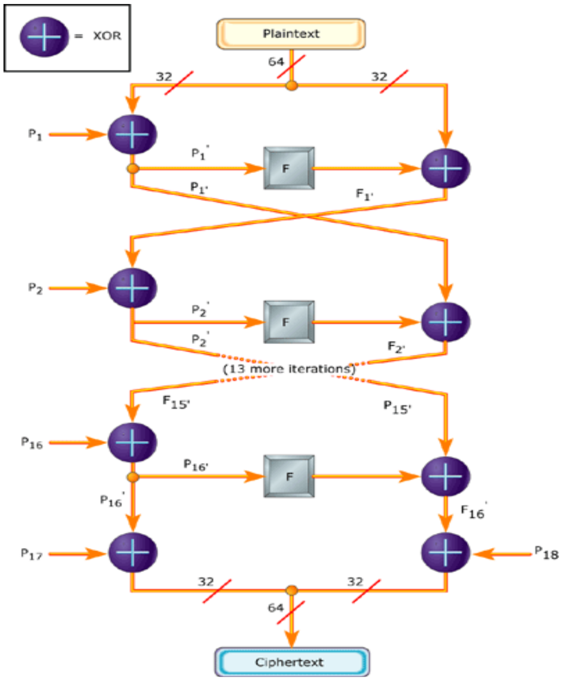

Figure.4: Blowfish Encryption Algorithm

IV. DESIGN OF ENHANCED BLOW-FISH ALGORITHM

As shown in fig.5 below, Blowfish algorithm is divided in to two parts: Encryption & Decryption unit for data processing and Sub-key generation Unit for generation of sub-keys to be used in each round of operation. In the data encryption and Decryption block, input 64-bit data block is divided in to two halves as 32-bit Left Encryption (LE) and 32-bit Right Encryption (RE). In each round of operation, the algorithm will perform RE and LE operations as shown in equation (1) for encryption and equation (2) for Decryption which is also shown in fig.1 for Encryption. The Fiestel function (F) in each round consists of combination of substitution, addition/modulo addition, XOR and addition/modulo addition operations. Thus, the algorithm follows the procedure for 16-rounds. RE16 and LE16 are XORed with P17 and P18 respectively to generate RE17 and LE17. Reverse operation is performed for the decryption operation [1, 2, 7].

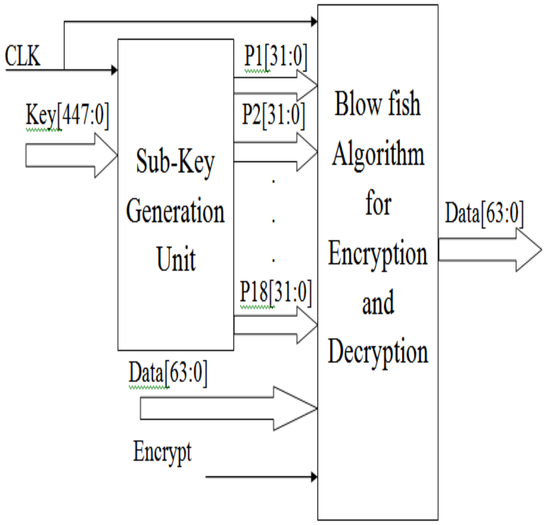

Figure.5: Top level Design module of Blowfish Algorithm

V. RESULTS AND DISCUSIONS

Enhanced Blowfish algorithm is developed in Verilog HDL and implemented it on ModelSim front end tool with Altera 6.3g P1 (Quartus- II 8.1) Web Edition and Xilinx ISE Design Suite 14.2. The implementation of AES Algorithm is compared with Enhanced Blowfish Algorithm. The implementation of the design followed bottom up approach. The test bench is written in Verilog HDL for every module of the design to provide 100% code coverage of the design. Top level Test Bench (TB) of the design is instantiated with top module of the design which consists of all the sub modules instantiated in it. Test cases are generated, applied to the Design under Test (DUT) and results are generated for further verification of functionality, Delay estimation, frequency of the design and Throughput calculation. As it is reflected in the results shown in table 1 and fig.6, Enhanced Blowfish algorithm has less encryption and decryption delay, increased frequency. Thus the total delay is decreased and frequency is increased compared to AES algorithm. In this design, parallelism in implementation decreased delay and thus increased the frequency.

TABLE1: COMPARISONS OF BLOWFISH AND AES ALGORITHMS FOR ENCRYPTION AND DECRYPTION DELAY AND FREQUENCY

S. NO.	Crypto processor algorithm/ parameter	Encryption Delay(ns)	Encryption Frequency (MHz)	Decryption Delay(ns)	Decryption Frequency (MHz)	Total Delay (ns)	Total Frequency (MHz)
1	Enhanced Blowfish	70.08	14.26	70.08	14.26	71.067	14.07
2	AES	77.66	12.87	77.66	12.87	158.93	6.29

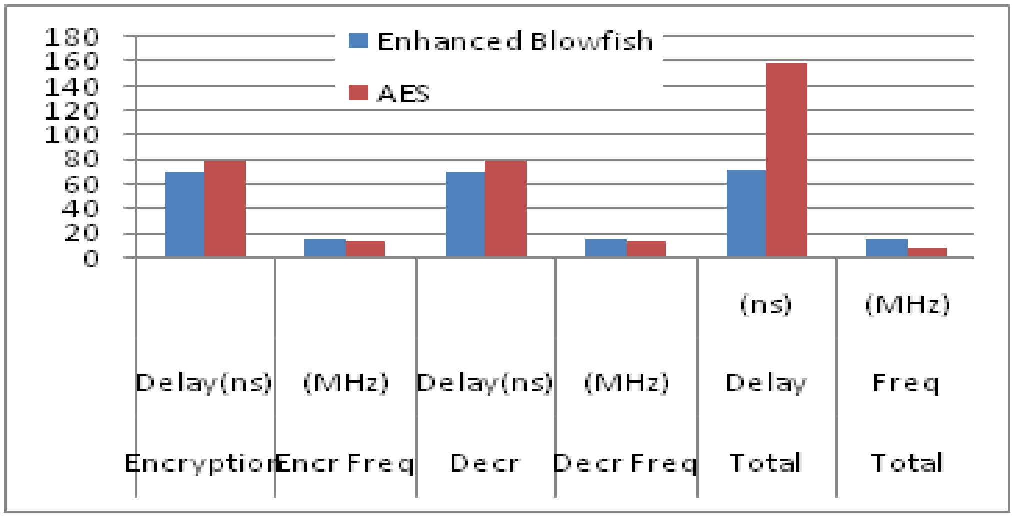

FIGURE.6: COMPARISONS OF BLOWFISH AND AES ALGORITHMS FOR ENCRYPTION AND DECRYPTION DELAY AND FREQUENCY WITH BAR CHART

As shown in table 2, the enhanced blowfish has less delay and more through put is more because of improved architectural design of enhanced Blowfish Algorithm compared to AES.

As depicted below in figure 7, bar chart graph shows that Enhanced Blowfish shows improvement over AES algorithm in all the parameters considered except memory utilization.

VI. CONCLUSIONS

As discussed in the results and discussion that Enhanced Blowfish Algorithm implementation gave better results compared to other implementations.

Constant delay n-bit adder circuit used in the Blowfish Algorithm which reduced the delay to 71.067ns, increased frequency to 14.07MHz and thus increased throughput to 840Mbps compared to AES implementation. It is providing more security because of 448 bit key length and incorporating WDDL logic in the Encryption and

Decryption process of Crypto-processor digital design flow. Future scope of this research work is to decrease the delay, improve the frequency and yielding better throughput with improved architectural design.