

An Enhanced Approach to Improve the Performance of Blowfish Algorithm by using 512 KB ROM

Sravana Sandya Gopavarapu¹, Lavanya Nalluri²

¹ PG Scholar, Dept of ECE, QIS Institute of Technology, Ongole, AP, India.

² Assistant Professor, QIS Institute of Technology, Ongole, AP, India.

Abstract:

It is well-known that advanced encryption standard (AES) algorithm is used for protection against various classes of wireless attacks in wireless communication standard such as Wi-Fi, Wi-MAX, Zig-bee and Bluetooth. However, the AES is a complex algorithm that consumes a larger design core, time, and power source. Hence, this paper presents a development of an improved power-throughput Blowfish algorithm on Zynq-7000 field-programmable gate array (FPGA) as an alternative security algorithm. In the extension for this work we proposed memory-based method is used to optimize the performance of Blowfish by replace 1024 ROM with the 512 kb of ROM due to that The performance is analyzed in terms of its architecture, throughput, and power consumption.

Keywords

advanced encryption standard, Blowfish, security, power-throughput, field-programmable gate array.

I. Introduction

Cryptography is a science of information security. It is the art of protecting the data. It stores and transmits the information safely over the insecure medium like Internet by encoding text data into a form non recognizable format with the help of various encryption algorithms and only the intended user will be able to convert it into original text. The process which converts original data into the unreadable form is called encryption process. The

encrypted data is called cipher text. The reverse of data encryption is data decryption which converts the cipher text back into the original text. Original text is also called plain text. Cryptology is a combination of Cryptography (encryption) and cryptanalysis (decryption).

Cryptography algorithms are classified as:

- Symmetric (private key) algorithm.
- Asymmetric (public key) algorithm.

In symmetric algorithms uses only one key for encrypt the data and same for decrypt the data. Asymmetric key algorithm uses two keys, one is used to encrypt the data and other is used to decrypt the data. Length of Key has an important place in Symmetric key encryption. For the same algorithm, encryption using longer key is hard to cryptanalyze means more secure as compared to the one using shorter key.

Asymmetric encryption techniques are almost one-thousand times slower than symmetric techniques as they require more computational processing power. Blowfish has been examined for five years. Serge Vaudenay has examined weak keys in Blowfish. Vincent Rijmen's Ph.D. paper includes a second-order differential attack on 4-round Blowfish. The key of the Blowfish algorithm is 448 bits, so it re-quires 2^{448} combinations to examine all keys.

Bruce Schneier designed blowfish in 1993 as a fast, free alternative to existing encryption algorithms. Since then it has been analyzed considerably, and it is slowly gaining acceptance as a strong encryption algorithm. The Blowfish algorithm has many advantages. It is suitable and efficient for hardware implementation and no license is required. The elementary operators of Blowfish algorithm include table lookup, addition and XOR. The table includes four Sboxes and a P-array. Blowfish is a cipher based on Feistel rounds, and the design of the F-function used amounts to a simplification of the principles used in DES to provide the same security with greater speed and efficiency in software. Blowfish is a 64 bit block cipher and is suggested as a replacement for DES. Blowfish is a fast algorithm and can encrypt data on 32-bit microprocessors.

The Blowfish algorithm has many advantages. It is suitable and efficient for hardware implementation. Besides, it is unpatented and no license is required.

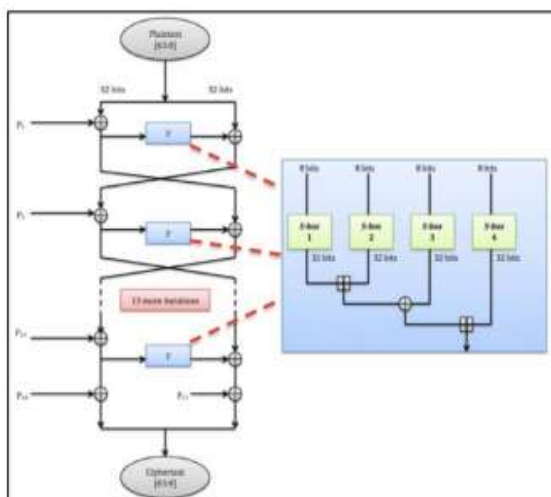


Fig 1. Blowfish Algorithm with F Function

The proposed Blowfish is designed using a memory-based method to improve its performance. This design is extensively evaluated based on three

areas. The first area is the architectural parameter, which is used to obtain a minimum hardware requirement that can lead to a smaller design size. The second area is a high-throughput design to carry out an encryption/decryption as fast as possible. Finally, the third area is the low power design, which seeks to minimize power consumption at all costs. This comparison can help researchers decide on the possibility of implementing Blowfish for a secure wireless communication instead of AES.

This paper is organized as follows. Section 2 discusses the related works on Blowfish designs. Section 3 introduces the improved power-throughput Blowfish architecture. Section 4 analyzes the performance of the proposed Blowfish in terms of architecture, throughput, and power consumption. Finally, Section 5 presents the conclusion.

II. Related Works

This section discusses the related studies on Blowfish designs. These works are presented to show the performance comparison in terms of architecture, throughput, and power consumption using FPGA. Not all studies on the Blowfish algorithm were designed with very high-speed integrated circuit hardware description language (VHDL) or Verilog, whereby the design can be simulated and then implemented on FPGA for verification. Most studies analyzed performance based on simulation by only using Matlab, CPU processor, and schematic, which will not be discussed in this section.

A soft-core implementation of the Blowfish cryptographic algorithm known as SCOB, was proposed by Salomao et al. [9]. This soft-core processor is oriented toward applications that demand a high throughput and exploit both the spatial and temporal parallelism available in the

Blowfish algorithm [9]. A VHDL was used to design this Blowfish and implemented on Altera Flex10K epf10k250agc599-1 FPGA. The design utilizes six random access memory (RAM) and was run at 10 MHz of clock frequency. Singpeil et al. [10] proposed an implementation of the Blowfish algorithm in the commercial FPGA coprocessor micro Enable to obtain a high performance design. The speed of Blowfish computation was increased by a factor of 10. The operating frequency for this Blowfish was 10 MHz

A VHDL model of Blowfish was designed by Raghuram et al. [11]. its architecture consists of addition, subtraction, modular multiplication, exponentiation, and XOR units. A high data rate is achieved by applying loop unrolling to the Montgomery algorithm [11]. The maximum clock frequency of this design is 77 MHz Sudarshan et al. [12] proposed an architecture using dynamic reconfiguration, replication, inner loop pipelining, and loop folding techniques. It was implemented on Virtex2 2v500fg456-6 FPGA with a clock frequency of 146.515 MHz another Blowfish algorithm was developed and implemented on Virtex xcv50bg256-6 FPGA by Karthigai Kumar and Baskaran [13]. The iterative method was used in their work to reduce the occupied area. Only a single register was required instead of a huge number of registers, which gives feedback to itself for each round. Thus, four memory ports for four S-boxes and one memory port for Parry are needed for their design. The Blowfish design was run with a clock frequency of 95 MHz.

Dakate and Dubey [14] designed a Blowfish with a 128-bit key size using VHDL, and it was implemented on Altera Quartus II FPGA. The key generation of using the F function was proposed. However, their study did not discuss the clock

frequency and design size of their Blowfish algorithm. The latest Blowfish design on FPGA was presented by Chatterjee et al. [15]. The algorithm was developed using Verilog and verified through Spartan3E xc3s500e-5fg320. They used a pipelined approach to design the algorithm to improve its throughput. Their architecture showed that the data path of sixteen module blocks is measured by a control unit. Their Blowfish design was operated at 295.63 MHz with a latency of 49 clock cycles.

III. Proposed

Blowfish In this study, an improved power-throughput Blowfish algorithm was designed using Verilog. The architecture of the proposed Blowfish consists of a 128-bit block size and key size, whereby it comprises two parallel blocks of 64-bit Blowfish algorithm that are simultaneously executed. This design technique enables the throughput of the Blowfish algorithm to be maximized. As shown in Fig. 2, the parallel blocks share the same S-box that is used for the F function. On Spartan 3E FPGA, BRAM is utilized to store the four 32-bit S-boxes where the performance can be improved by decreasing the delay into the clock-to-out value of the flip-flop (FF) [16]. The mode is used to select for encryption or decryption.

As the implementation of the Blowfish design is targeted to reduce the core size and timing delay, the proposed memory based S-box method is optimized as illustrated in Fig. 3(a). Based on the Verilog design module, In the existing method uses a read-only memory (ROM) that contains 1024×32 -bit input data of addr. In the proposed method we reduce the read-only memory (ROM) that contains 512×32 -bit input data of addr, to decrease the area and delay of the existing method. The addr represents the data of four 32-bit Sboxes with 256 entries each.

The 32-bit output data are read from the ROM at a positive clock edge. The proposed method can also lessen the total of slices used by the Blowfish design. A slice contains a set number of look-up tables (LUTs), FFs, and multiplexers. Thus, less logic resources are used to perform logic, arithmetic, and ROM functions that can lead to a faster encryption/decryption process [16].

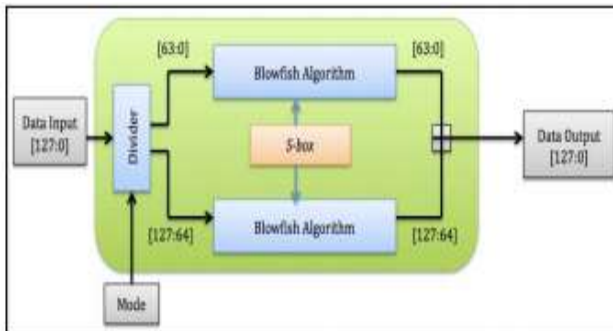
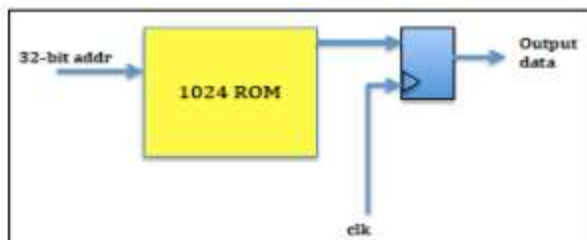
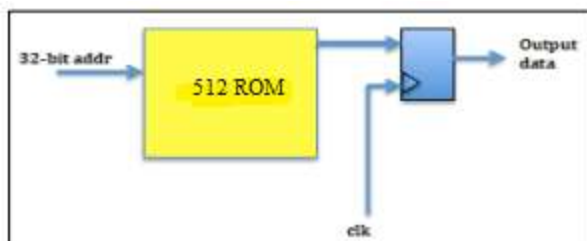


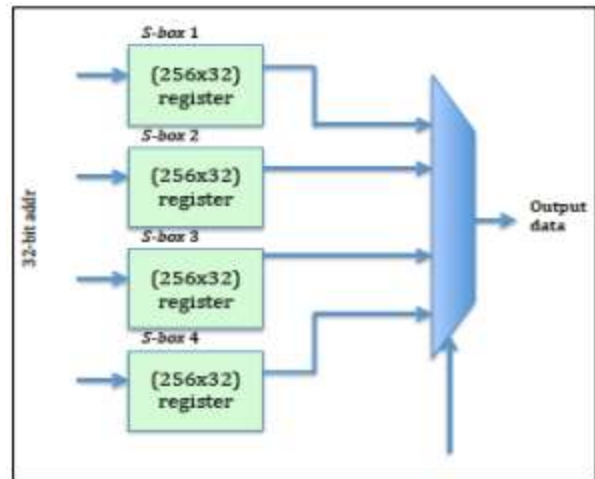
Fig. 2. Block Diagram of the Proposed Blowfish Design.



(a) Existing Memory-based Method



(b) Extension Memory-based Method



(c) Design Method by [12]

Fig. 3. Schematic Comparison of Different Methods for S-boxes.

The proposed method is compared with the S-box method invented by [12] as shown in Fig. 3(b), since their output results are the best among others. The 256×32 -bit registers are used for each S-boxes, which means the involvement of many groups of FFs that store a bit pattern. A single register has a clock, input data, and output data, and it enables a signal port. The 32-bit input data of addr are latched and stored internally for every clock cycle. This method can slow down the speed of the Blowfish performance as each register has its own timing delay.

IV. Throughput

Throughput is defined as the average rate of successful message delivery over a communication channel. In this paper, throughput is directed toward evaluating each architecture's characteristic and performance. Throughput is calculated as Eq. 1 based on [18].

$$\text{Throughput (Gbps)} = 128 \text{ bits} * \text{Clock Frequency (MHz)} / \text{Latency} \quad (1)$$

Latency is the encryption/decryption time that is calculated in clock cycles. Latency should be as small as possible to achieve a power-saving system. Furthermore, a long battery life is necessary, particularly for mobile devices.

For the proposed Blowfish, the achieved throughput is 2183 Mbps, which is 29% higher than the output result in [12]. The latency for each encryption and decryption mode is 19 clock cycles. Figure 4 shows the performance comparison between the proposed Blowfish design and previous studies. The proposed Blowfish is clearly the fastest with the smallest design size.

V. EXISTING RESULTS

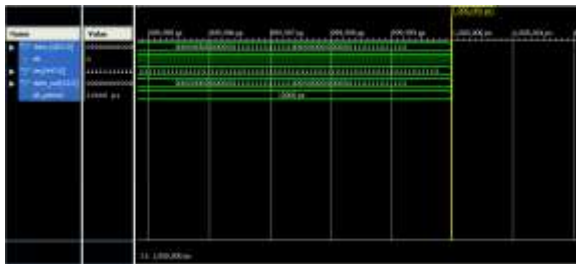


Fig 4: Outputs using Existing Method

VI. PROPOSED RESULTS

Timing summary

Minimum period: 3.101ns (Maximum Frequency: 322.435MHz)

Minimum input arrival time before clock: 3.597ns

Maximum output required time after clock: 2.199ns

Maximum combinational path delay: 2.361ns

Process "Synthesize - XST" completed successfully

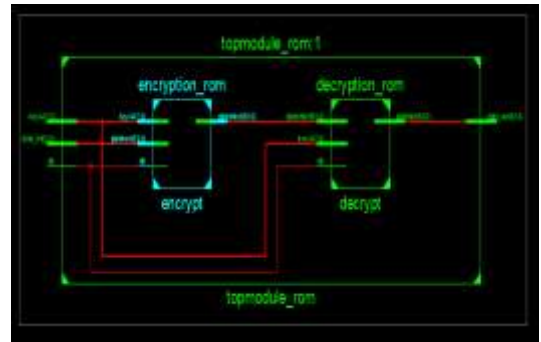


Fig (a): RTL diagram layout

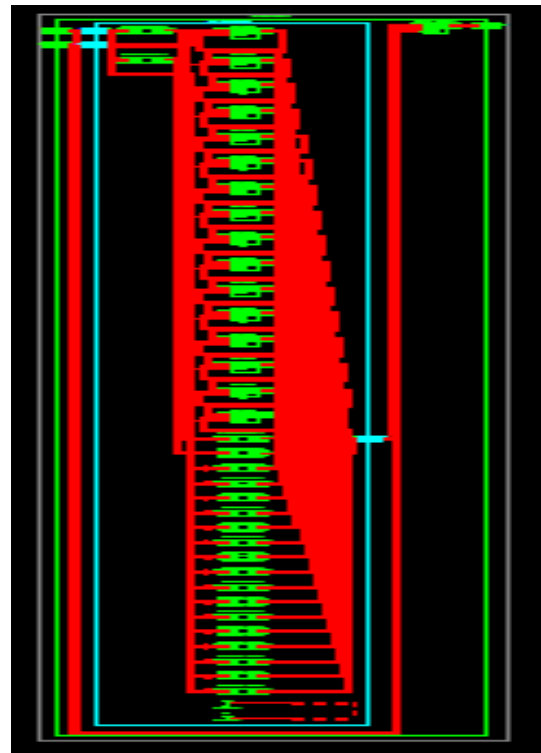


Fig (b): RTL diagram Internal

Fig 5: RTL Diagram

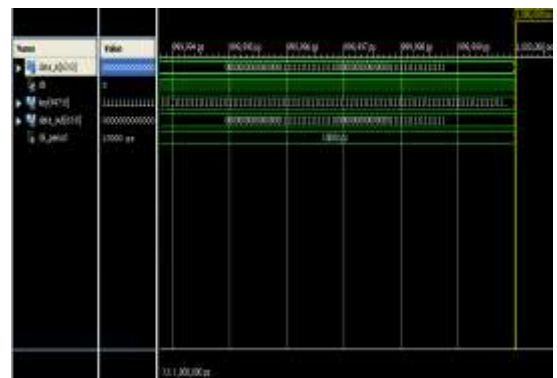


Fig 6: Extension output wave forms

Table 1: Device utilisation summary

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	50	35200	0%
Number of Slice LUTs	74	17600	0%
Number of Fully used LUT-FF pairs	38	86	44%
Number of Bonded IOBs	39	100	39%
Number of BUFG/BUFGCTRLs	1	32	3%



Fig 7: Simulation Results

VII. Conclusion and Future Scope

This paper presents a development of an improved power throughput Blowfish as a security algorithm, which is best embedded in mobile devices for wireless communication. Seven papers from previous studies are compared with the proposed Blowfish, and the performance is verified through reprogrammable FPGA. The optimal performance is defined strictly by the least number of hardware requirements, highest throughput, and lowest power consumption. The output results presented in this paper indicate that the proposed Blowfish has the smallest design core and highest throughput, with low power consumption. These findings prove the superiority of the proposed Blowfish design. These characteristics are necessary for current research trends given that the data need to be transmitted with

a high speed using a low power source for energy efficiency.

References

- [1] D. S. Abd Elminaam, H. M. A. Kader, and M. M. Hadhoud, "Evaluating the performance of symmetric encryption algorithms," International Journal of Network Security, vol. 10(3), pp. 213-219, 2010.
- [2] J. Thakur, and N. Kumar, "DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis," International Journal of Emerging Technology and Advanced Engineering, vol. 1(2), pp. 6- 12, 2011.
- [3] D. K. Dakate, and P. Dubey, "Performance comparison of symmetric data encryption techniques," International Journal of Advanced Research in Computer Engineering & Technology, vol. 1(4), pp. 163-166, 2012.
- [4] A. Kumar, and S. Karthikeyan, "Investigating the efficiency of Blowfish and Rejindael (AES) algorithms," International Journal Computer Network and Information Security, vol. 2, pp. 22-28, 2012.
- [5] P. C. Mandal, "Superiority of Blowfish algorithm," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2(9), pp. 196-201, 2012.
- [6] S. Pavithra, and E. Ramadevi, "Study and performance analysis of cryptography algorithms," International Journal of Advanced Research in Computer Engineering & Technology, vol. 1(5), pp. 82-86, 2012.

- [7] M. Mathur, and A. Kesarwani, "Comparison between DES, 3DES, RC2, RC6, Blowfish and AES," in Proceedings of National Conference on New Horizons in IT (NCNHIT2013). Mumbai, India, 2013, pp. 143-148.
- [8] B. Schneier, and D. Whiting, "Fast software encryption: Designing encryption for optimal speed on the Intel Pentium processor," in Proceedings of 4th International Workshop on Fast Software Encryption, LNCS Springer Verlag, 1997, pp. 242-259.
- [9] S. L. C. Salomao, J. M. S. de Alcantara, V. C. Alves, and A. C. C. Vieira, "SCOB, a softcore for the Blowfish cryptography algorithm," in Proceedings of the IEEE International Conference on Integrated Circuit and System Design, 1999, pp. 220-223.
- [10] H. Singpiel, H. Simmler, A. Kugel, R. Manner, A. C. C. Vieira, F. Galvez-Durand, J. M. S. de Alcantara, and V. C. Alves, "Implementation of cryptographic applications on the reconfigurable FPGA coprocessor microEnable," in Proceedings of the 13th Symposium on Integrated Circuits and Systems Design. Manaus, Brazil, 2000, pp. 359-362.
- [11] S. Sukumar, Raghuram, Chaitali, and Chakrabarti, "Programmable processor for cryptography," in Proceedings of IEEE International Symposium on Circuits and Systems. Geneva, Switzerland, 2000, pp. 685-688.
- [12] T. S. B. Sudarshan, R. A. Mir, and S. Vijayalakshmi, "DRIL-a flexible architecture for Blowfish algorithm encryption using dynamic reconfiguration, replication, inner-loop pipelining, loop folding techniques," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 3740 LCNS, 2005, pp. 625-639.
- [13] P. Karthigaikumar, and K. Baskaran, "Partially pipelined VLSI implementation of Blowfish encryption/decryption algorithm," International Journal of Image and Graphics, vol. 10(3), pp. 327-341, 2010.
- [14] D. K. Dakate, and P. Dubey, "Blowfish encryption: A comparative analysis using VHDL," International of Engineering and Advanced Technology (IJEAT), vol. 1(5), pp. 177-179, 2012.
- [15] S. R. Chatterjee, S. Majumder, and B. Pramanik, "FPGA implementation of pipelined Blowfish algorithm," in Proceedings of 5th International Symposium Electronic System Design. Mangalore, India, 2014, pp. 208-209.
- [16] Xilinx, Virtex6 Family Overview, Product Specification, DS150, v2.4. USA, 2012, pp. 1-2.
- [17] Xilinx, Zynq-7000 All Programmable SoC Overview, Product Specification, DS190, v1.8. USA, 2015, pp. 1-14.
- [18] A. J. Elbirt, W. Yip, B. Chetwynd, and C. Paar, "An FPGA implementation and performance evaluation of the AES block cipher candidate algorithm finalists," in Proceedings of the Third Advanced Encryption Standard Candidate Conference, National Institute of Standards and Technology (NIST). New York, 2000, pp. 13-27.
- [19] Xilinx, Power Methodology Guide, UG786, v13.1. USA, 2011, pp. 8-9.