Implementation of a High-Performance Blowfish for Secure Wireless Communication

Rafidah Ahmad[1], Asrulnizam Abd. Manaf[1], Widad Ismail[2]
[1]*Collaborative MicroElectronic Design Excellence Centre (CEDEC), Sains@USM, Level 1, Block C,No. 10 Persiaran Bukit Jambul, 11900 Penang, Malaysia.*
[2]*Auto-ID Laboratory, School of Electrical & Electronic Engineering, Universiti Sains Malaysia, Engineering Campus, 14300 Nibong Tebal, Penang, Malaysia.*
rafidah.ahmad@usm.my

***Abstract*—Security has become an important concern in wireless communication systems both for the users and the providers. Without a secure medium, users can have their transmissions eavesdropped or sensitive information tapped. Therefore, this paper focuses on the implementation of security algorithm, which is best embedded in the mobile devices. It is well known that most wireless communication standards implement an advanced encryption standard (AES) as a security algorithm. However, the AES is a complex algorithm that consumes a larger design core, time, and power source. Hence, a high-performance Blowfish algorithm is proposed and developed using a Zynq-7000 field-programmable gate array (FPGA). The performance is analyzed in terms of its architecture, throughput, and power consumption. Results show that the proposed Blowfish reduces slice usage by 63% and increases throughput by 29% at low power consumption.**

***Index Terms*— Blowfish; Security; Wireless Communication.**

I. INTRODUCTION

There are many types of wireless communication standard such as WiFi, WiMAX, Zigbee and Bluetooth. A mobile network environment basically consists of mobile station, cell antenna, base station, mobile switching centre, location register and authentication server [1]. Different from wired networks, where the physical transmission medium can be secured, wireless networks use the air as a transmission medium [2]. The air interface is exposed to both active and passive attacks. Since all transmitted data travel directly between a mobile devices and the base station, the possibility to copy the data transmitted through the air is very high. Hence, a device level security algorithm is needed as a protection against various wireless attacks [3]. Although few researchers claimed that software implementations of security algorithm are considered to provide ease of use, upgrade, portability and flexibility, the implementations of security algorithm in hardware are preferred because they are faster than software implementations and more physically secure, where key access and algorithm modification are harder [4].

Mobile devices are designed to be portable with small and light characteristics. The power consumption of mobile devices indicates the battery performance. In order to conserve the energy, the complexity of the security algorithm needs to be reduced with less processing time. Nowadays, numerous security algorithms are available and used for information security across insecure networks. The IEEE standard has incorporated the advanced encryption standard (AES) algorithm to provide strong data encryption for various types of wireless communication standards.

However, by referring to a study investigated by [5–11] on the performance comparison between AES and Blowfish, the result shows that the AES actually consumes more power and time than Blowfish. Blowfish was designed in 1993 by Bruce Schneier as a free and simple alternative to existing security algorithms. Blowfish has a 64-bit block size and a variable key length from 32 bits to 448 bits [12]. The Blowfish algorithm consists of two units: key expansion and data encryption units. Figure 1 shows that the 64-bit text input is divided into two 32-bit halves in this algorithm. Blowfish uses P-array (P1-P18), which consists of 18 32-bit subkeys for key expansion unit, and has 16 rounds, with each round implementing the Feistal (F) function. In the F function block, four 32-bit S-boxes have 256 entries each. After the 16th round, two 32-bit halves data are recombined to obtain the cipher text.

This paper proposes the implementation of high-performance Blowfish algorithm on a Zynq-7000 xc7z020 field-programmable gate array (FPGA) platform. FPGA is used for the implementation process because it can be reconfigured for multiple tasks with only a single chip [13]. FPGA platforms also support for high processing speed and low power consumption, where they are perfect as a prototype of mobile devices. The proposed Blowfish is designed using a memory-based method to improve its performance. This design is extensively evaluated based on three parameters. The first parameter is the architectural parameter, which is used to obtain a minimum hardware requirement that can lead to a smaller design size [13]. The second parameter is a high-throughput design to carry out an encryption/decryption as fast as possible [13]. Finally, the third parameter is the low power design, which seeks to minimize power consumption at all costs [13]. 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 proposed high-performance 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.

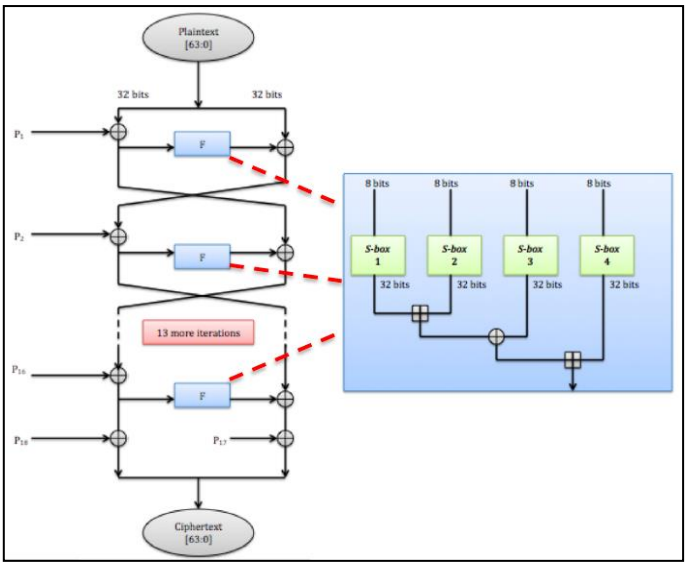


Figure 1: Blowfish algorithm with F function

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. [14]. 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 [14]. A VHDL was used to design this Blowfish and implemented on Altera Flex19K epf10k250agc599-1 FPGA. The design utilizes six random access memory (RAM) and was run at 10 MHz of clock frequency. Singpeil et al. [15] proposed an implementation of the Blowfish algorithm in the commercial FPGA coprocessor microEnable 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. [16]. 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 [16]. The maximum clock frequency of this design is 77 MHz. Sudarshan et al. [17] 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 Karthigaikumar and Baskaran [4]. 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 P-array are needed for their design. The Blowfish design was run with a clock frequency of 95 MHz.

Dakate and Dubey [18] 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. [19]. 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.

Table 1 summarizes the performance analysis conducted by [4, 14–19] in terms of architecture, throughput, and power consumption. The optimal performance of the Blowfish design was obtained by [17] with the smallest design core of 65% and the highest throughput of 1545.7 Mbps. Meanwhile, the lowest power consumed is 66 mW by [4] with a throughput of 780 Mbps. However, the slices used are 93%, which is the largest compared with other studies. The highest power consumption was discovered by [18] with a difference of 87%, as compared with [4]. These findings also prove that a low throughput indicates a slow timing process and high power consumption.

Table 1
Performance comparison on Blowfish designs based on previous studies

Reference	Design Size (slices)	Throughput (Mbps)	Power Consumption (mW)
[4]	1608/1728 (93%)	780.0	66
[14]	-	40.0	-
[15]	-	38.0	-
[16]	-	44.0	-
[17]	65%	1545.7	-
[18]	-	303.6	515
[19]	3222/4656 (69%)	386.1	-

III. PROPOSED HIGH-PERFORMANCE BLOWFISH

In this study, a high-performance 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 Figure 2, the parallel blocks share the same S-box that is used for the F function. On Zynq-7000 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) [20]. The mode is used to select for encryption or decryption.

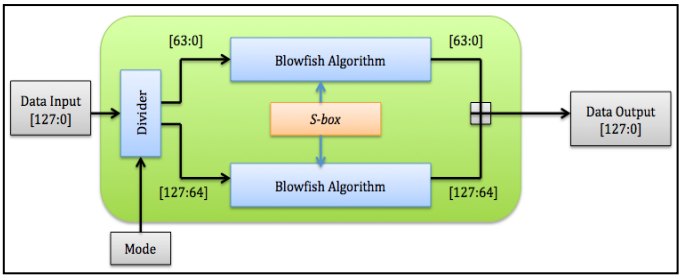


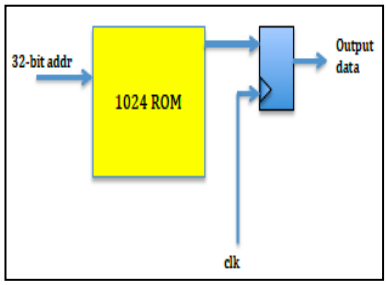
Figure 2: Block diagram of the proposed Blowfish design

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 Figure 3(a). Based on the Verilog design module, the proposed method uses a read-only memory (ROM) that contains 1024 × 32-bit input data of *addr*. The *addr* represents the data of four 32-bit *S-boxes* 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 [20].

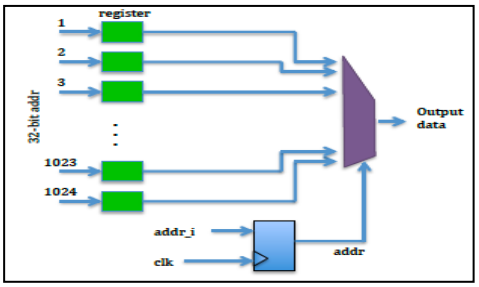
However, based on the conventional method invented by [12] as shown in Figure 3(b), 1024 × 32-bit registers are used for four 32-bit *S-boxes*, which means the involvement of many groups of FFs that store a bit pattern. A single register has a clock, input data, and output data, and it enables a signal port. The 32-bit input data of *addr* are latched and stored internally for every clock cycle, and the output data are updated based on a positive edge of *clk* and *addr_i* to match the internally stored data. This method can slow down the speed of the Blowfish performance as each register has its own timing delay.

IV. COMPARISON OF RESULTS

In this section, the performance of the proposed high-performance Blowfish architecture is compared with the architectures from previous studies. This section is divided into three parts: architectural characteristic, throughput, and power consumption. The maximum clock frequency for the proposed architecture is 324 MHz.



(a)



(b)

Figure 3: Schematic comparison of different methods for S-boxes. a. Proposed memory-based method. b. Conventional method

A. *Architectural Characteristic*

Table 2 shows the hardware requirement of the proposed Blowfish based on a post-implementation report. The proposed Blowfish requires only 2% of the slices and FFs compared with [4, 17, 19]. For Zynq-7000, four LUTs and eight FFs, as well as multiplexers and arithmetic carrying logic, form a slice [21]. Two slices form a configurable logic block. One FF per LUT can optionally be configured as latches [21]. The proposed Blowfish also uses 3% LUTs, which is used for random logic implementation or distributed memory. The usage of BRAM that consists of S-boxes is about 5% with 16 input/output (IO) pins. These results show that the proposed Blowfish design has the smallest design core with the least number of hardware requirements.

Table 2
Architectural characteristic of the proposed Blowfish design

Slices	FF	LUT	BRAM	IO
635/26600 (2%)	2539/106400 (2%)	1849/53200 (3%)	7.5/140 (5%)	16/200 (8%)

B. *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 [22].

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

Latency is the encryption or decryption time that is calculated in clock cycles. In reliable two-way communication systems, latency limits the maximum rate that information can be transmitted. If a security algorithm is directed toward a device that wakes up, captures data, encrypts data, transmits data, and reverts to sleep mode, latency can become an issue because the longer the system needs to be awake, the more power is required [23]. Hence, 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 [17]. 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.

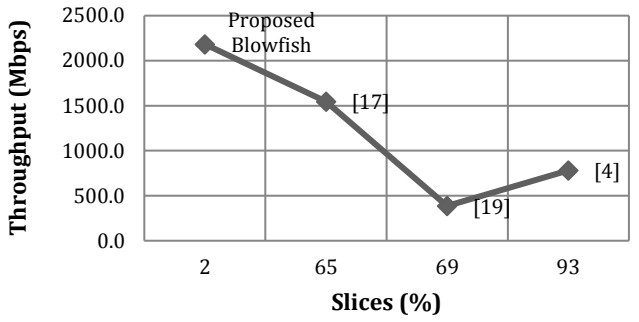


Figure 4: Performance comparison of the proposed Blowfish with previous studies

C. Power Consumption

The power requirement for the proposed Blowfish architecture is discussed in this section. The Xilinx XPower analysis tool is used to analyze power consumption. In this paper, the dynamic power for both architectures is compared. Dynamic power is associated with design activity and switching events in the core or IO of the device [24]. Dynamic power is determined in terms of clock trees, logic, signal, BRAM, and IO power. This analysis introduces a highly efficient method of locating the blocks or parts of the design that are the most deprived in the power aspect, thereby providing an easy path to power optimization [24].

Table 3 shows the power consumption performed by the proposed Blowfish design. From this table, the power consumed in BRAM is 21%, which means that the four 32-bit S-boxes consume only low power during its operation. Meanwhile, logic power is the lowest with a total of 10%. This finding indicates that the proposed Blowfish uses few logic functions that can lead to rapid encryption and decryption processes. Given that this Blowfish is operated at a frequency of 324 MHz, the total power consumed is 142 mW, which is higher than [4]. However, [4] was run at a low frequency of 95 MHz. Therefore, through FPGA, a higher clock frequency will consume more power for the design operation.

Table 3
Power consumed by the proposed Blowfish design at 324 MHz operating frequency

Power (mW)					
Clock	Signals	Logic	BRAM	IO	Total
34	21	14	30	43	142

V. CONCLUSION

Security has become a very serious concern in a wireless communication system. Security is critical to protect the availability, integrity, and privacy of all connected resources and information. This paper presents a high-performance Blowfish as a security algorithm, which is best embedded in mobile devices for wireless communication standards. 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.

ACKNOWLEDGMENTS

This research was financially supported by the Ministry of Science, Technology, and Innovation (MOSTI) ScienceFund Research Grant (Project No. 06-01-05-SF0640) and the Universiti Sains Malaysia Research University Grant for Individuals (RUI) (Project No. 1001/PELECT/814241).

REFERENCES

[1] Kumar A.: Mobile Broadcasting with WiMAX. 1st edn. ScienceDirect, New York (2008)
[2] Imai H., Rahman M.G., Kobara K.: Wireless Communications Security (Artech House Universal Personal Communications). 1st edn. Artech House Publishers, Norwood, MA (2006)
[3] Ahmad R., Ismail W.: A Survey of High Performance Cryptography Algorithms for WiMAX Applications Using SDR. In: Al-Dulaimi, A., Cosmas, J., Mohammed A. (eds) Self-organization and green applications in cognitive radio networks. Chapter 11, pp. 231-246. IGI-Global, USA (2013)
[4] Karthigaikumar, P., Baskaran, K.: Partially Pipelined VLSI Implementation Of Blowfish Encryption/Decryption Algorithm. International Journal of Image and Graphics. 10(3), 327-341 (2010)
[5] Abd Elminaam D.S., Kader, H.M.A., Hadhoud, M.M.: Evaluating the Performance of Symmetric Encryption Algorithms. International Journal of Network Security. 10(3), 213-219 (2010)
[6] Thakur, J., Kumar, N.: DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis. International Journal of Emerging Technology and Advanced Engineering. 1(2), 6-12 (2011)
[7] Dakate, D.K., Dubey, P.: Performance Comparison of Symmetric Data Encryption Techniques. International Journal of Advanced Research in Computer Engineering & Technology. 1(4), 163-166 (2012)
[8] Kumar, A., Karthikeyan, S.: Investigating the Efficiency of Blowfish and Rejindael (AES) Algorithms. International Journal Computer Network and Information Security. 2, 22-28 (2012)
[9] Mandal, P.C.: Superiority of Blowfish Algorithm. International Journal of Advanced Research in Computer Science and Software Engineering. 2(9), 196-201 (2012)
[10] Pavithra, S., Ramadevi, E.: Study and Performance Analysis of Cryptography Algorithms. International Journal of Advanced Research in Computer Engineering & Technology. 1(5), 82-86 (2012)
[11] Mathur, M., Kesarwani, A.: Comparison Between DES, 3DES, RC2, RC6, Blowfish and AES. In: Proceedings of National Conference on New Horizons in IT (NCNHIT2013), pp. 143-148. Mumbai, India, 2013
[12] Schneier, B., Whiting, D.: Fast Software Encryption: Designing Encryption for Optimal Speed on the Intel Pentium Processor. In: Proceedings of 4th International Workshop on Fast Software Encryption. LNCS, pp. 242-259. Springer Verlag (1997)
[13] Ahmad, R., Ismail, W.: Performance Comparison of Advanced Encryption Standard-128 Algorithms for WIMAX Application with Improved Power-Throughput. Journal of Engineering Science and Technology. 11(12), 1-17 (2016)
[14] Salomao, S.L.C., de Alcantara, J.M.S., Alves, V.C., Vieira, A.C.C.: SCOB, A Softcore For The Blowfish Cryptography Algorithm. In: Proceedings of the IEEE Internatonal Conference on Integrated Circuit and System Design, pp. 220-223, 1999
[15] Singpiel, H. Simmler, H., Kugel, A., Manner, R., Vieira, A.C.C., Galvez-Durand, F., de Alcantara, J.M.S., Alves, V.C.: Implementation of

Cryptographic Applications On The Reconfigurable FPGA Coprocessor MicroEnable. In: Proceedings of the 13th Symposium on Integrated Circuits and Systems Design, pp. 359-362, 2000

[16] Sukumar, S., Raghuram, Chaitali, Chakrabarti.: Programmable processor for cryptography. In: Proceedings of IEEE International Symposium on Circuits and Systems 5, pp. 685-688, 2000

[17] Sudarshan, T.S.B., Mir, R.A., Vijayalakshmi, S.: 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). LCNS, pp. 625-639. Springer Verlag (2005)

[18] Dakate, D.K., Dubey, P.: Blowfish Encryption: A Comparative Analysis Using VHDL. International of Engineering and Advanced Technology (IJEAT). 1(5), 177-179 (2012)

[19] Chatterjee, S.R., Majumder, S., Pramanik, B.: FPGA Implementation Of Pipelined Blowfish Algorithm. In: Proceedings of 5th International Symposium Electronic Sytem Design, pp. 208-209, 2014.

[20] Xilinx: Virtex6 Family Overview. Product Specification, DS150, v2.4, USA (2012)

[21] Xilinx: Zynq-7000 All Programmable SoC Overview. Product Specification, DS190, v1.8, USA (2015)

[22] Elbirt, A.J., Yip, W., Chetwynd, B., Paar, C.: 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), pp. 13-27, New York, USA (2000)

[23] Dyken, J., Delgado-Frias, J.G.: FPGA Schemes For Minimizing The Power-Throughput Trade-Off In Executing The Advanced Encryption Standard Algorithm. Journal of Systems Architecture. 1-8 (2010)

[24] Xilinx: Power Methodology Guide, UG786, v13.1, USA (2011)