International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
International Conference on Industrial Automation and Computing (ICIAC-12-13th April 2014)

RESEARCH ARTICLE OPEN ACCESS

Design of enhanced speed Blowfish Algorithm for cryptography with merged encryption & decryption in VHDL

Mr.Tushar Joshi[1], Mr. Ravindra Yadav [2], Mr. Utsav Malviya[3]
Student, M-Tech.VLSI & embedded, GGITS, Jabalpur 2. Asst.Professor ITM College of Engineering, Kamptee, 3.Asst.Professor,GGITS,Jabalpur

Abstract:
Data security has always been important in all aspects of life. Data may contain several form of information that we want to secure from any unauthorized access. It can be all the more important as technology continues to control various operations in our day to day life. Reprogrammable devices are highly attractive options for hardware implementations of encryption algorithms as they provide cryptographic algorithm agility, physical security, and potentially much higher performance, therefore this paper investigates a hardware design to efficiently implement a special type block ciphers in VHDL and its comparative analysis in different parameter variation . This hardware design is applied to the new secret and variable size key block cipher called Blowfish designed to meet the requirements of the previous known standard and to increase security and to improve performance. The proposed algorithm will be used a variable key size.

There has been a tremendous enhancement in the field of cryptography, which tries to manipulate the plaintext so that it becomes unreadable, less prone to hacker and crackers, and again obtain the plaintext back by manipulating this unreadable text in some way. In this regard, we have modified secure algorithms which are secret key block ciphers that enhance performance by modifying their function. We have shown that total time taken for encryption and decryption is reduced for both the algorithms after the modification. We have also made an attempt to show that this improvement will not violate the security when compared to that of existing Blowfish algorithm. Because the change in the total time taken for encryption and decryption cannot be understood on software implementation, we have implemented VHDL application to show the differences in the delay.

Keywords: Blowfish, encryption, decryption, feistel network, subkeys.

I. Introduction

Blowfish is a variable-length key, 64-bit block cipher. The algorithm consists of two parts: a key expansion part and a data- encryption part. Key expansion converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes. Data encryption occurs via a 16-round Feistel network as shown in Figure 2.Each round consists of a keydependent permutation, a key and data-dependent substitution. All operations are EX-ORs and additions on 32-bit words.

The Blowfish encryption scheme was designed by Bruce Schneider in 1993 to replace Data Encryption Standard (DES), which was the Federal Information Processing Standard Cryptography (FIPS Crypto) . The intent was to create a cryptographic algorithm which did not possess the limitations and issues common in other crypto algorithms and to provide an open, readily available crypto for users rather than the common patented or classified crypto algorithms being used. Due to its standing as a crypto algorithm. Blowfish continues to attain its lofty goals of secure, open encryption that is realizable in software and hardware.

The Blowfish algorithm is conceptually simple, but its actual implementation and use is complex. Blowfish has a fixed 64-bit block size. The key length of Blowfish is anywhere from 32 bits to 448 bits. The cipher is a 16-round Feistel network and uses password-dependent S-boxes. A Feistel network is one that utilizes a structure which makes encryption and decryption very similar through the use of the following elements:

P-boxes (permutation boxes, these perform bit shuffling)
S-boxes (substitution boxes, simple nonlinear Functions)
XORing to achieve Linear Mixing

Blowfish encapsulates all these elements into an efficient and powerful algorithm. The action of Blowfish can be seen in Figure2.

II. Blowfish Algorithm

Data encryption occurs via a 16-round Feistel network. Each round consists of a key-dependent permutation, and a

key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data look-ups per round. Blowfish uses a large number of sub-keys. These keys must be precomputed before any data encryption or decryption. The key array also called P-array consists of 18 32-bit subkeys: P1, P2,...,P18.

There are four 32-bit S-boxes with 256 entries each:
S1,0, S1,1,..., S1,255;
S2,0, S2,1,..., S2,255;
S3,0, S3,1,..., S3,255;
S4,0, S4,1,..., S4,255.

Encryption: Blowfish is a Feistel network consisting of 16
rounds as in Figure 1. The input is a 64- bit data element, x.
Divide x into two 32-bit halves: xL, xR
For i= 1 to 16:
xL = xL XOR Pi
xR = F(xL) XOR xR
Swap xL and xR
Swap xL and xR (Undo the last swap.)
xR= xR XOR PI7
xL = xL XOR P 18
Recombine xL and xR
Function F:
Divide xL into four eight-bit quarters: a, b, c, and d
F(xL) = ((SI,a + S2,b mod 232) XOR S3,c) + S4,d mod 232

Decryption is exactly the same as encryption, except that PI,
P2 ... P 18 are used in the reverse order. Implementations of
Blowfish that require the fastest speeds should unroll the loop and ensure that all sub keys are stored in cache.

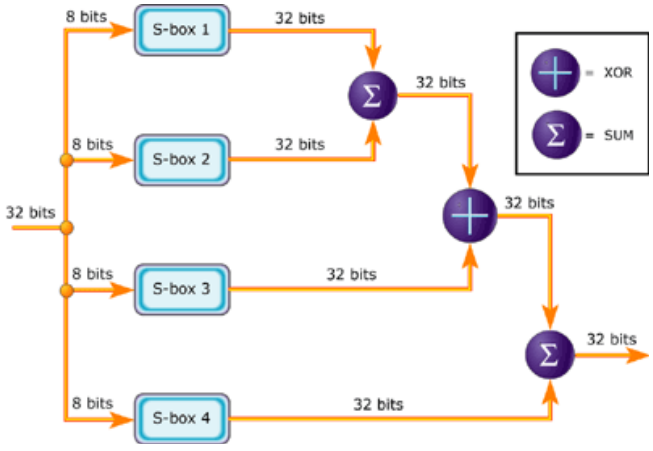

Figure 1: Graphic representation of F

III.HARDWAREIMPLEMENTATION USING VHDL

VHDL (Very High Speed Integrated Circuit Hardware Description Language) was chosen as a language used to describe the improvement suggested to algorithm stated above. VHDL has many features appropriate for describing the behavior of electronic components ranging from simple logic gates to complete microprocessors and custom chips. Features of VHDL allow electrical aspects of circuit behavior (such as rise and fall times of signals, delays through gates, and functional operation) to be precisely described.

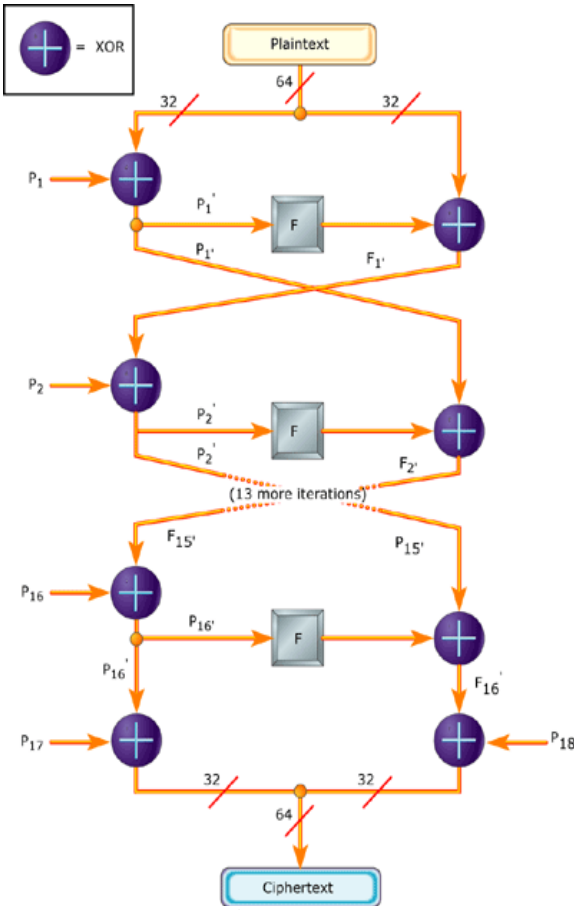

Figure 2: Blowfish algorithm

This paper also is presenting a VHDL architecture (structural) in order to write the design code of the encryption algorithm that describe improved blowfish algorithm. The presented architecture allows keeping the flexibility of the algorithm by taking advantage of generic VHDL coding. It executes one round per clock cycle, computes the round and the key round in parallel and supports both encryption and decryption at a minimal cost.

IV.ENCRYPTION DEVELOPMENT

As to develop the logic, key generation is one of the complex schedules of Blowfish encryption

because of the fact that it includes variable size key. So for the development of the key the use of S BOX (Substitution Box) can be a greater aid for developing of logic of this algorithm.

Figure 1.shows the key generation concept of this particular implementation using VHDL. As we provide 8 bit input to single S Box, which outputs 32 bits data, similarly in case of rest of S Boxes the same concept to be followed. After which addition and XOR operation generates the key which is used for data encryption.

After the initial proposal of the Blowfish cipher, Dr. Dobb's journal sponsored a cryptanalysis contest in order to ascertain the security of Blowfish. The high hopes and large promises provided by the algorithm yielded much activity regarding a potential cryptanalysis. Given this fact, it is not surprising that many interesting results were proposed, ever, none came close to actually successfully cracking or providing a cryptanalysis of Blowfish. Some of the most intriguing results of the last century will be presented here for completeness. Only five results in total were submitted. John Kesley could only break 3-round Blowfish (nowhere near the 16-round final version), and his cryptanalysis cannot be extended beyond 3 rounds. Serge Vaudenay used an intentionally weakened version of Blowfish and found a known-plaintext attack requiring 28r + 1 (where *r* is the number of rounds) known plaintexts to break; however, this method is impractical in reality and does not work against the full 16-round Blowfish algorithm. The most promising attack was proposed in 1996 by Vincent Rijmen in his doctoral dissertation, but this attack can only break 4 rounds of Blowfish and no more. The most recent work is from Dieter Schmidt, who noted that the third and fourth sub keys are independent from the user's 64-bit key .Given that these attempts are the only ones known thus far and that they are surprisingly weak in decrypting the actual Blowfish algorithm, the future of Blowfish as a secure algorithm is very promising indeed.

Table 1: Block and Stream Cipher Speed Comparison

Algorithm	**Type**	**Clocks/ Round**	**#Round**	**Clock/Byte of output**
RC4	Stream cipher	n. a	n. a	7
SEAL	Stream cipher	n. a.	n. a	4
Blowfish	Block Cipher	9	16	18
RCS	Block Cipher	12	16	23
DES	Block Cipher	18	16	45
IDEA	Block Cipher	50	8	50
Triples-DES	Block Cipher	18	48	108

V. PROPOSED DESIGN

Our design utilizes the simplicity of Blowfish to create a relatively straightforward implementation. The S-box contains a table of non-linear data which maps inputs to
outputs. This design severely disrupts any sort of correlation between the input and the output of the design. When the Blowfish design is started, the S-boxes and the P-boxes are initialized based on the input key and nonlinear data. The writing of S-box data is based on linear equation and the user which can be varied in accordance with necessity of security which is the slowest part of the design. The table lookup procedure is handled without a clock pulse, so a uniform clock is possible between initialization and runtime without sacrificing performance. The design uses a structural style of modeling to initialize and perform its encryption/decryption functionality. The first state initializes the internal state counter, effectively preparing it for use in the next phase and throughout the encryption/decryption process. The second state initializes the P functions with nonlinear data derived from PI. This is an important step in assuring proper encryption, as the digits of PI have no currently known cryptographically exploitable pattern. The counter is incremented 1024 times as the P functions get XOR ed against the key. This step prepares the Blowfish algorithm for encryption and decryption by properly preparing the system for use. The next state rewrites the S-boxes with new information based on the key. A ready state is now entered where the system waits for an encryption or decryption signal from the user. Upon receiving this signal, it enters either an encryption or a decryption state which lasts 17-18 cycles. This design is compact, efficient, and very fast. This can be seen in the schematic below, showing the final overall design. The desired result as far as a size profile is clearly achieved.

VI. RESULTS

The Blowfish implementation proposed here has been implemented, and several important test vectors were encrypted and decrypted to guarantee correct function. Our finalized Blowfish cipher implementation can perform block cipher encryption and decryption in variable frequency of input. Using the compile-ultra switch, the maximum clockspeed is 167 MHz (a 6 ns clock). The throughput of the design is 3.68069 Gbits/sec for encryption, and 559 for decryption. Power consumption is variable on the input data frequency, and area is 4608 cells. A comparison of our design with the leading publicly available *high-speed* Blowfish implementations is shown in Table 2.

Architeture	**Throughput**	**Size**	**Clock Frequency**	**Process (TSMC)**
Salomao et al	266 Mbits/sec	4620 std. cells	66 Mhz	0.7
Lin et al	200 Mbits/sec	16k	50 Mhz	0.6
Lin et al	288 Mbits/sec	13k	72 Mhz	0.35
Our design	**3.68069 Gbits/sec**	**4996 cells**	**--------**	**0.35**

Table 2: Comparison of High Speed Blowfish Hardware implementation

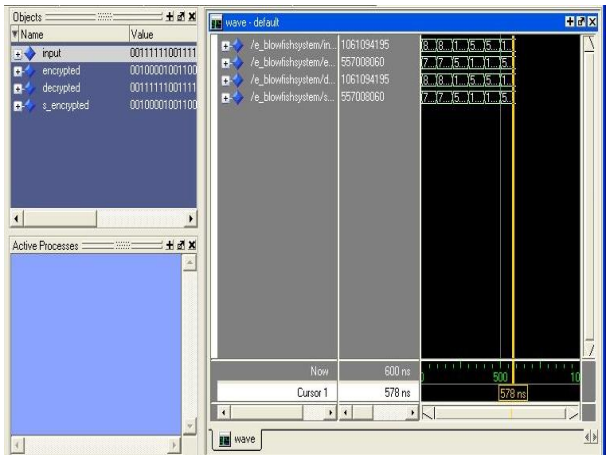

Figure 3: Encryption and Decryption of a Test Vector

V. CONCLUSION

A high-speed implementation of the Blowfish cryptographic algorithm has been presented. Even without pipelining, which other high-speed methods utilize, our method was rated at 3.68069 Gbits/sec maximal throughput, which is more fast as the leading (pipelined) competitor. With pipelining, our design could reach 10.667 Gbits/sec throughput. The overall resulting design from our scheme was considerably large but only consumed variable power on input data frequency during operation.

Our solution takes advantage of the conceptual simplicity of Blowfish and is optimized for high-speed encryption and decryption. The results show that our design objective is achieved, especially when compared to the leading competitors.

REFERENCES

[1] High Speed SOC Design for Blowfish Cryptographic Algorithm Brian Cody1 Justin Madigan2 Spencer MacDonald3 Kenneth W. Hsu4* 1, Brian.j.cody@gmail.com, Kulicke & Soffa Industries 2, Justin.madigan@gmail.com,3, som0749@gamil.com, DAE Systems, 4, kwheec@rit.edu, Rochester Institute of Technology * Correspondence author *2007 IFIP International Conference on Very Large Scale Integration (VLSI-SoC 2007)*

[2] B. Schneier, Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish),Fast Software Encryption, Cambridge Security Workshop Proceedings (December 1993), Springer-Verlag, 1994.

[3] M. Thaduri, S.-M. Yoo, An efficient VLSI implementation of IDEA encryption algorithm using VHDL, Sciencedirect ,5 JUNE 2004.

[4] Afaf M. Ali Al-Neaimi, New Approach for Modifying Blowfish Algorithm by Using Multiple Keys, IJCSNS, March 2011.

[5] Krishnamurthy G.N , V. Ramaswamy , Leela G.H and Ashalatha M.E," Blow-CAST-Fish: A New 64-bit Block Cipher", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.4, p. 282,April 2008.

[6] P. Karthigai Kumar, K. Baskaran , An ASIC implementation of low power and high throughput blowfish crypto algorithm, Science direct, 6 April 2010.

[6] Sushanta Kumar Sahu, Manoranjan Pradhan, FPGA Implementation of RSA Encryption System, International Journal of Computer Applications, April 2011.

[7] B. Schneier, "Applied Cryptography," 2nd ed. New York: , JohnWiley & Sons, Inc., 1996.

[8] B. Schneier. The Blowfish Encryption Algorithm. Retrieved 12:04:58, July 27, 2007 from http://www.schneier.com/blowfish.html

[9] B. Schneier, "Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish)," *Fast Software Encryption: Second International Workshop, Leuven, Belgium, December 1994, Proceedings*, Springer-Verlag, 1994, pp. 191-204.

[10] B. Schneier, Speed Comparisons of Block Ciphers on a Pentium. Retrieved 12:04:58, July 27, 2007 from http://www.schneier.com/blowfish-speed.html.