AN IMPLEMENTATION OF DATA ENCRYPTION FOR INTERNET OF THINGS USING BLOWFISH ALGORITHM ON FPGA

[1] Kurniawan Nur Prasetyo ST.
Departement of Computer System, School of Engineering Telkom University, Bandung, Indonesia
nurprasetyo.kurniawan@gmail.com

[2]Yudha Purwanto, ST., MT.
Departement of Computer System, School of Engineering Telkom University, Bandung, Indonesia
omyudha@telkomuniversity.ac.id

[3] Denny Darlis, S.Si., MT.
Diploma of Telecommunication Engineering, School of Applied Science, Telkom University, Bandung, Indonesia
dennydarlis@telkomuniversity.ac.id

Abstract

Information security has become an important issue in data communications. One method to ensure the security of data is to use cryptographic method. Cryptography is a method to encode the information to keep the information from being hacked by the other party. The implementation of cryptography is used a significant amount of computer resources. Various range application of blowfish algorithm can be implemented for data encryption sent from an Internet of Things physical network which have IP-based data. In this research, blowfish algorithm is implemented on FPGA using VHDL programming language, and monitored the number of FPGA resource that is used. The blowfish algorithm is analyzed by computing certain metrics performances such as security, encryption time, avalanche effect, and throughput from multiple testing scenarios for system reliability. The testing showed that blowfish algorithm gave a good performance when implemented in FPGA and show a good alternative to proposed as network security on Internet of Things.

Keywords : Internet of Things, Encryption, Blowfish, VHDL, FPGA.

1. Introduction

The encryption algorithm consists of two models, symmetric key encryption algorithm and asymmetric key encryption algorithm. Symmetric key encryption algorithm or private encryption is encrypted using the same key for encryption and decryption. Security of symmetric key depends on the secrecy of the key. While the asymmetric encryption algorithm is an algorithm that uses different keys for encryption and decryption. One example of the symmetric encryption algorithm is blowfish. Blowfish algorithm has the simple structure, it can perform encryption and decryption process quickly, has not yet being patent and have a high level of security.

There are two ways to implement the cryptographic algorithms which are via software or hardware implementation. One type of hardware implementation is using hard-wired components, for example FPGA. VHDL language is used for implementation on FPGA. This research is used FPGA for the implementation because of several reasons. FPGA is cheap, easy to implement, reprogrammed, has high speed and has a good level of security.

This research is focused on performances and implementation of blowfish algorithm. The performance measure of encryption algorithm schemes conducted changing round fiestel and changing key size. The performances parameters that were discussed are encryption time, FPGA implementation resource used, avalanche effect, and throughput.

This paper is consisted of 5 sections. Section 1 is the introduction parts. Section 2 presents the related work of this research. Section 3 describes blowfish algorithm design. The implementation, result and analysis are presented in section 4. And the last section 5 is the conclusion of the research and implementation.

2. Related Work

To give more description about the performance of the blowfish algorithm, this section shows some other works from related field.

In [13], authors analyzed the performance of DES, AES and Blowfish encryption algorithms. Their performances were compared by varying block size, key size and number of round of the encryption input file. The performances are analyzed by computing certain performance parameters such as execution time, memory required, and throughput. The result showed blowfish algorithm consumes less execution time, memory usage and produces more throughputs. Blowfish performed approximately 4 times faster than AES and 2 times faster than DES. AES showed poor performance results compared to other algorithms, since it required more power for processing.

A study in [12] is conducted to analyze popular secret key algorithms such as DES, 3DES, and Blowfish. Those algorithms were implemented, and compared by varying content and size of the encryption input files. The algorithms were tested on two different hardware platforms, P-II 266 MHz and P-IV 2,4 GHz. The results showed blowfish has very good performance compared to the other algorithms. It also showed that 3DES has 1/3 throughput of DES, and also needed thrice longer time than DES to process same amount of data.

978-1-4799-3580-2/14/$31.00 ©2014 IEEE

In [1] study provided evaluation of common encryption algorithms such as, AES, DES, 3DES, RC2, Blowfish, and RC6. There were some basic parameters of performance such as battery power consumption, encryption or decryption speed compared. The results showed that blowfish had better performance than other algorithms when changing packet size. 3DES still had low performance compared to DES algorithms. RC2 showed the poorest performance among all.

In [9] the blowfish encryption scheme has been studied, and its effectiveness demonstrated by implementing a windows file encryption tool. The secrecy of the cipher has also been discussed, and it was shown that so long as the plaintext alphabet of characters is relatively evenly distributed and the key has chosen at random, little knowledge of the key or plaintext can be gain from the ciphertext. Although some attempt have been made at attacking blowfish, significant weaknesses have yet to be shown when the full 448 bit key and 16 rounds are used. Therefore, it is still a very effective cipher to be used by a common computing user to encrypt a set of sensitive files.

From [7] author improved blowfish algorithm by modifying its function. The paper showed performance of analyzed blowfish and compare it to its modified version. The results conducted the security of the modified algorithm is at least as good as that of original blowfish algorithm. It showed that the proposed algorithm has better avalanche effect than any other of the existing algorithm and hence can be incorporated in the process of encryption of any plaintext. It also showed that ancient cipher has very less avalanche effect and hence cannot be used for encryption.

In [14] author designed algorithm that combines the process scrambling of bits from ancient cipher and substitution boxes from modern cipher. It implemented the use caesar cipher and vigenere cioher of ancient cipher and combined it with DES and blowfish.

In [8], the authors modified two secure algorithms which are Blowfish and CAST-128. Author modified those algorithms in secret function. The total time taken for encryption and decryption from modified algorithms and the original were compared. The algorithms were implemented in VHDL application to show the differences in the delay. The result showed that the modified algorithm was faster than the original one.

In [2] study showed how the modified blowfish algorithm is implemented on FPGA. Authors proposed various modifications in blowfish algorithm. The results showed that modified blowfish algorithm provides a low power implementation since the processing paths can be executed in parallel speed which can be done on FPGA using a very fast speed.

In [3], it was the study of modified blowfish algorithm implemented on FPGA. There are three changes proposed which are key expansion, extension to 128 bit, and modification of F-function. The result showed that FPGA implementation of modified blowfish algorithm provides a low power implementation.

This research designed blowfish algorithm using VHDL language, analyzed the performance, and finally implemented it on FPGA module.

3. Design

Blowfish algorithm is a symmetric block cipher that can be effectively used for encryption and safeguarding of data. It consists of two parts:

- Key Expansion
  Change the function key (Minimum 32 - bit and maximum 448 - bit) into 18 entry P-array and four 256 entry S-Boxes.

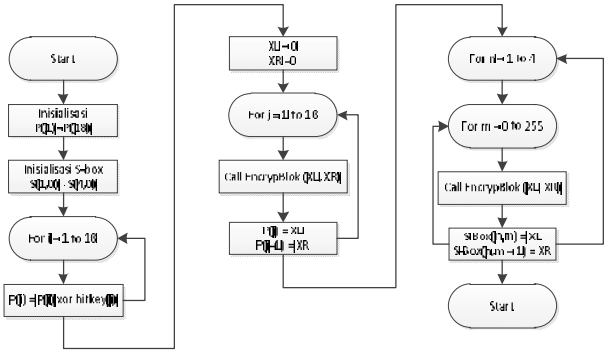

Figure 1: Key expansion process

- Data Encryption
  Iteration consists of 16 times round of simple functions (Feistel Network)

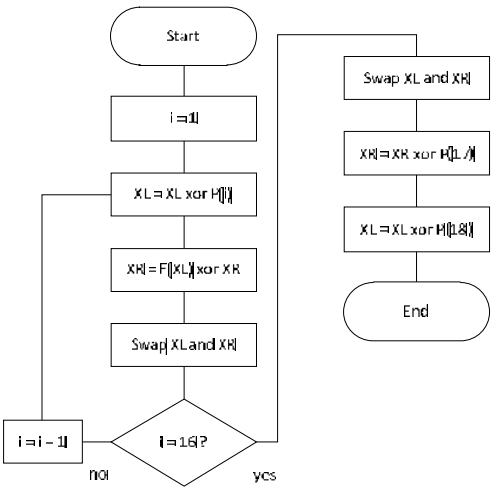

Figure 2: Encryption process

The verifying process of the system was done using ISim features on Xilinx. After verifying the system, the blowfish algorithm was implemented in FPGA Virtex-4 XC4VLX25 SF363.

Several performance parameters were analyzed which are encryption time, throughput, and FPGA resource used. The security parameter is calculated using avalanche effect. The encryption time is taken to produce a ciphertext from plaintext where ignored the time required to set up the key [12]. The throughput can

978-1-4799-3580-2/14/$31.00 ©2014 IEEE

be calculated as the total plaintext in bytes divided by the encryption time encryption time [1]. FPGA resources were the amount of slices, LUTs, and IOB of FPGA used to implement blowfish algorithm.

There are two scenarios performed and one implementation analysis in this paper.

- The original blowfish algorithm used 16 rounds of fiestel. But in this paper, the number of rounds fiestel was reduced to 8 rounds and 4 rounds. This research parameters concerned encryption time, throughput and avalanche effect.
- The key size was changed from 448 bit to 384 bit, 320 bit, 256 bit, 192 bit, 128 bit and 64 bit. This researched analyzed the FPGA resources used and avalanche effect.
- The implementation and analysis of blowfish algorithm on FPGA used Virtex-4 XC4VLX25 SF363.

4. Result

4.1. Reduces Round Fiestel

4.1.1. Encryption time

Encryption time is time that used for producing cyphertext from plaintext, regardless time used for key generation. Value of P-array and S-boxes are fixed in the iteration and has been set before the simulation. Before encryption process, reset has been enabled and hold for 2 clock cycles, while the clock changed every 5 ns.

Table 1: Encryption Time Table

Process	16	8	4
Reset	25 ns	25 ns	25 ns
Inisialisasi proses XOR	30 ns	30 ns	30 ns
Roundfiestel	160 ns	80 ns	40 ns
Total Encryption Time	215 ns	135 ns	95 ns

Table 1 shows that, less number of rounds will reduces total encryption time. From the table above, it can be concluded once round of fiestel needs 10 ns or one cycle clock.

4.1.2. Throughput

Throughput is the amount of information or data flowing from one place to the other place. Throughput of encryption scheme can be calculated by dividing the total plaintexts in Mbytes on the total encryption time for each encryption process.

$$Throughput : \frac{Plaintext\ (Mbyte)}{Encryption\ time\ (secon)}$$

Table 2: Throughput comparison (in Mbytes/sec)

Round	4	8	16
Throughput	84,2105	59,2592	37,2093

From the table above, it showed that the lesser number of round fiestel in blowfish algorithm the greater throughput was accomplished. It means that the faster encryption time will produce more throughputs.

4.1.3. Avalanche Effect

A change in one bit of the plaintext or one bit of the key should produce a change many bits of the cyphertext. This change in number of bits in the cyphertext whenever there is a change in one bit of plaintext or one bit of key is called avalanche effect [8].

Table 3: Avalanche effect comparison

Number of round	4	8	16
Avalanche Effect	48,29%	50,1%	49,04%

In this research, there are 32 samples of plaintext with same key. Changing the plaintext by one bit between 32 samples. And the results shows in table 3.

4.2. Changing key Size

4.2.1. FPGA Resources needed

For implementation into FPGA modules, we should pay attention in the number of resources that are available on the FPGA, because FPGA has limited resources.

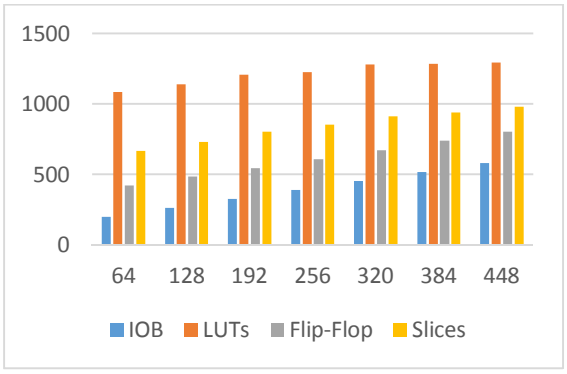

Figure3: FPGA resources needed

IOB is pin resource needed to implement the design in FPGA. In this research, the design takes 64 plaintext inputs, 64 ciphertext outputs, and 9 control pins. 137 pins are needed if we ignore input key. Meanwhile the number of pin is available in FPGA Virtex-4 XC4VLX25 SF363 were 240 pins. So the key length must be less than 103 pins, if we want to implement it in FPGA. Therefore only 64 bit key lengths that can be implemented in FPGA. LUTs (Look Up Tables) is the implementation of the syste, that was designed before. LUTs consist of several inputs and one output, where the output can be programmed for each input according the design. Flip-flop is temporary register, flip-flop stores input value in semi-permanent form until there is a command to delete or change the bit stored. Slices are

978-1-4799-3580-2/14/$31.00 ©2014 IEEE

the combination of LUTs and Flip-flop. From the figure above, it can be concluded that the larger key length is the more resources is needed to implementation in FPGA.

4.2.2. Avalanche Effect

In this research, avalanche effect was calculated by number of different ciphertext from the plaintext in various key lengths. There are 32 samples for each key length. It has the same plaintext, but with different size and value of the key length. Avalanche Effect refers to a desirable property of cryptographic algorithms where, if an input is changed slightly the output changes significantly.

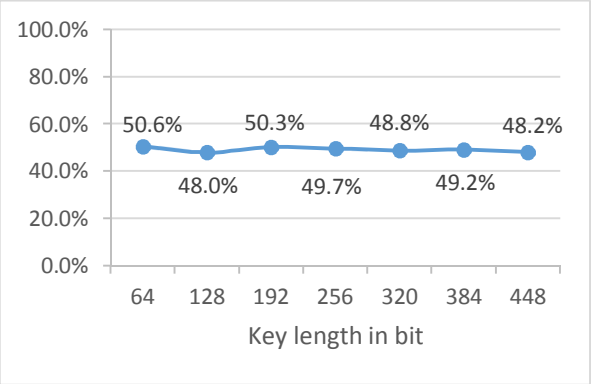

Figure 4: Avalanche effect comparison

From the figure, it shows that the difference of key length does not effect in avalanche effect. The average avalanche effect is about 50%.

4.3. FPGA Implementation

The designed Blowfish algorithm was be implemented on FPGA Virtex-4 XC4VLX25 SF363. Xilinx Chipscope is used for viewing the output of the blowfish algorithm or ciphertext. FPGA source used for the implementation is shown on table 4.

Table 5 showed that the ciphertext output from chipscope is the same with the output from Isim. So it can be concluded that blowfish algorithm have a good performance when it is implemented on FPGA Virtex-4 XC4VLX25-SF363.

Table 4: FPGA resources needed

Source FPGA	Needed	Percentage
Slices	678	6%
Slices Flip-flop	424	1%
4 input LUTs	1024	5%
Bounded IOB	201	83%
FIFO 16	7	9%

This research uses 8 samples with different plaintext and key. Output ciphertext of the sample are compared the using chipscope and Isim.

Table 5: Ciphertext comparison

Plaintext	Ciphertext from ISim	Ciphertext from Chipscope
FFFFFFFFFFFFFFFF	51FC2ADFBDF84DCA	51FC2ADFBDF84DCA
3000000000000000	11A6663FDB647E52	11A6663FDB647E52
1111111111111111	A6113DFCB4FC6EBD	A6113DFCB4FC6EBD
0123456789ABCDEF	8989EA9B02A43633	8989EA9B02A43633
0000000000000000	4EF997456198DD78	4EF997456198DD78
FEDCBA9876543210	0D7B00C01A21E5C1	0D7B00C01A21E5C1

5. Conclusion and Future Work

This research presents the performance of blowfish algorithm with total time taken for encryption, avalanche effect and throughput from multiple testing scenarios as the parameters. The Blowfish algorithm was implemented on FPGA using VHDL language. The results shows that reducing the round of Fiestels reduce total encryption time, give greater throughput and not affect avalanche effect significantly. It also showed that larger key length needs more resources to implement on FPGA. Finally, blowfish algorithm is implemented on FPGA Virtex-4 XC4VLX25-SF363 well.

For the Future work, it is recommended that the input and plaintext is replaced using image or audio data as well as for the ciphertext. It is recommended to use FPGA board which better specification than Virtex-4 XC4VLX25 SF363. The modification of blowfish algorithm can be done to get better security or reduces total encryption time.

6. References

[1] A.E.Diaa Salama, Mohamed Abdual Kader.H, Mohamed Hadhoud.M,"*Evaluating The Performance of Symmetric Encryption Algorithms*", International Journal of Network security, PP.216-222,2010.

[2] Agrawal.M, Mishra Pradeep,"*A modified Approach for Symmetric Key Cryptography Based on Blowfish Algorithm*", International Journal of Engineering and Advanced Technology, ISSN : 2249-8958.2012.

[3] B.R.Akshtha, A.K.K.Amitabha, C.Neha, S.Jamuna, N.J.Raja."*FPGA Implementation of Modified Blowfish Algorithm*", International Conference on Electronics and Communication Engineering.2013.

[4] B.Schneier,"*Applied Cryptography 2nd Edition*", 1996.

[5] *B. Schneier,* "Description o f a New Variable Length Key, 64-Bit Block Cipher (Blowfish)", *Fast Software Encryption,Cambridge Security Workshop*

978-1-4799-3580-2/14/$31.00 ©2014 IEEE

roceedings (December1993), Springer-Verlag, 1994, pp. 191-204.

[6] B.Schneier, "*The Blowfish Algorithm – one year later*," Dr. Dobb's Journal, 1995.

[7] G.N.Krishnamurthy and Ramaswamy.V, "*Performance Analysis of Blowfish and its Modified Version using Encryption Quality, Key Sensitivity, Histogram and Correlation Coefficient Analysis*", International Journal of Recent Trends in Engineering,2009.

[8] G.N.Krishnamurthy, Ramaswamy.V, M.E.Ashalatha,"*Performance Enhancement of Blowfish and CAST-128 Algorithms and Security of Improved Blowfish Algorithms Using Avelanche Effect*",International Journal of Computer Science and Network Security,Vol.8 No.3, 2008.

[9] K.Mayers Rusell, H.Desoky Ahmed," *An Implementation of the Blowfish Cryptosystem*", IEEE.2008.

[10] L.P.Douglas,"*VHDL Programming by Example, 4TH Edition*", United State of America.2002.

[11] Munir, Rinaldi. 2006. *Kriptografi*. Bandung : Informatika Bandung.

[12] Nadeem.A, Younus Javed.M, "*A Performance Comparison of Data Encryption Algorithms*", IEEE,2005.

[13] Ramesh.A, Suruliandi.A, "*Performance Analysis of Encryption for Information Security*", IEEE, 2013.

[14] R.Sriram and K.Marimuthu,"*Designing an Algorithm with High Avelanche Effect*", International Journal of Computer Science and Network Security, Vol 11 No.1,2011.

[15] William Stallings, "*Cryptography and Network Security"*, Third Edition, Pearson Education, 2003.

978-1-4799-3580-2/14/$31.00 ©2014 IEEE