INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 3)
Available online at: www.ijariit.com

Implementation of modified blowfish algorithm

Bindu R.
bindu.gowda105@gmail.com
Presidency University, Bangalore, Karnataka

Dr. Deepak S. Sakkari
deepakssakkari@presidencyuniversity.in
Presidency University, Bangalore, Karnataka

ABSTRACT

This project focuses on implementing modified blowfish algorithm using a hardware description language such as VHDL. Encryption algorithm plays a major role in network application and data security systems. But securing data consumes a major amount of CPU time and battery power. We also focus on improvising the performance and security provided by the blowfish encryption algorithm. With advancement in technology, DES is found to be no longer secure. As a drop-in replacement for DES, blowfish encryption algorithm can be used. The original blowfish algorithm function has been modified using modified S-box and adding Key Bits Shifting (KBS) to the Function block.

Keywords— *Blowfish algorithm, Encryption, Security, Key Bits Shifting (KBS)*

1. INTRODUCTION

Cryptography is referred to as study of secret. This is a process where a readable message is converted into a form which is unreadable to others except for the one it is intended to. Whenever confidential information is sent, there is possibility of an unauthorized third-party attack in order to learn the confidential information [6].

Cryptography includes two basic components encryption algorithm and keys. If the sender and recipient use the same key then it is known as symmetric or private key cryptography. It is most suitable for long data streams. It is difficult to implement in practice as it is necessary for both sender and receiver to know the key. Moreover, the keys must be sent over a secure channel from sender to the receiver. The question behind is that if such a secure channel is already present, why not send the data directly over the secure channel. On the other hand, if different keys are used then it is known as asymmetric or public key cryptography. It is useful for short data streams.

Blowfish was designed by Bruce Schneider in 1993 as a fast, free alternative to existing encryption algorithms which was the Federal Information Processing Standard Cryptography (FIPS Crypto) [1] [2]. The algorithm is safe against unauthorized attack and runs faster than the popular existing algorithm.

The concept of blowfish is very simple to understand but its actual implementation and the use of algorithm in real time is very complex. Blowfish is a symmetric block cipher which can be used for encrypting and safeguarding the data effectively. Blowfish has a fixed 64-bit block size. Blowfish has variable length key, from 32 bits to 448 bits. Blowfish algorithm is a Feistel Network, iterating a simple encryption function 16 times. It consists of a complex initialization phase which is required before any encryption can take place. The actual encryption of data is very efficient on large microprocessors. As blowfish is a variable length key block cipher, it is most suitable for applications where the key does not often change such as a communications link or an automatic file encryptor [7].

1.1 Feistel Network

Horst Feistel published an article on Feistel Network in 1973. Most symmetric block ciphers use Feistel Network for their construction. A Feistel Network is said to be an iterative network which consists of an internal function called as round function. It is a method where the round function (also called as F-function) is transformed into permutation.

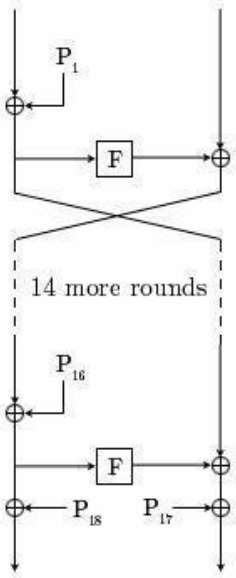

Fig. 1: Feistel Network

The Feistel Network working can be summarized as follows:

- The input data is split into two equal halves.
- The right half of input data becomes the new left half.

- The round function (F-function) is applied to the right half of input data and the key.
- The output of the F-function is then xored with the left half of input data.
- The output from the xor operation is the new right half.
- The new right half and the new left half becomes the Feistel Network output [7].

2. LITERATURE SURVEY OF BLOWFISH BASED ALGORITHMS

Blowfish algorithm as mentioned earlier is a 64-bit block cipher. It has variable length key varying from 32 to 448 bits. The algorithm consists of two parts: key expansion part and data encryption part. Key expansion part converts key which can be maximum of 448 bits into several sub-key arrays of total 4168 bytes [1].

Data encryption part is executed by a 16 round Feistel Network. In each round there is a key dependent permutation and key and data dependent substitution. The operations are usually xor and modulo addition on 32-bit words.

Additional operations include four indexed array data lookups per round. It has following elements:

- P-array (Permutation array which performs shuffling or mixing).
- S-boxes (Substitution boxes which performs nonlinear functions) [2].

2.1 Generating Round Keys and S-box

Generation of the round key is performed in rounds where each round generates two round key values. The process is as follows:

1. Initialize P and S-Boxes as described above
2. Exclusive-or P1 with the first 32 key bits, P2 with the next 32 bits and so on until all of the key has been exclusive-ored (since the key is shorter than P, parts of it will be used multiple times to cover all of P)
3. Set the initial input to zero
4. Encrypt the input using the current version of P as the round keys
5. Set the first two unreplaced values of P to the value of the cipher text from step 4
6. Set the input to the cipher text from step 4
7. Repeat steps 4 through 6 until all of P has been replaced
8. Use the resulting value of P as the round keys in encryption
9. Repeat steps 4 through 6, replacing values of the S-Boxes two at a time until all S-Box values have been replaced.

Since P contains 18 words and the S-Boxes each contain 256 words, there is a total of 18 + 4*256 = 1042 values to replace, which will take 521 iterations of steps 4 through 6 of the above algorithms to complete.

2.2 Implementation of S-BOX

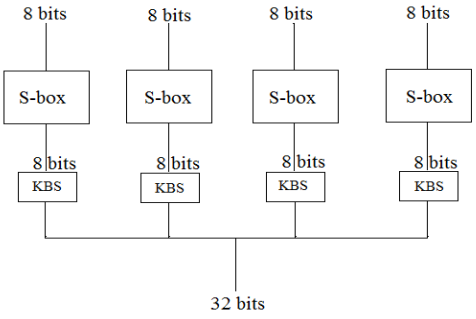

Fig. 2: Implemented S-BOX

We have modified the F-function of Blowfish by eliminating the modulo addition and replacing them by Key Bits Shifting (KBS).

The left side 32bits of data is divided into four 8bits of input for S-BOX where the operations takes place and encrypted output of 8bits is given to KBS that' converts plaintext to ciphertext.

The modified S-BOX function with the S-box and Mix column is implemented in the following manner:

1. The 32-bit input is divided into 4 S-bit quarters which are used as the input to the S-boxes.
2. The leftmost s-bits are sent to the first S-box and the output is given to KBS with the next 8-bit input. The output is given with 1 and given as the input to the second S-box
3. The outputs are XORed with the right side is divided 32bits the output plaintext. The output of step 3 is added to KBS with the last
4. The 32-bit input is divided into 4 S-bit quarters which are used as the input to the S-boxes.
5. The leftmost s-bits are sent to the first S-box and the output is given to KBS with the next 8-bit input. The output is given with 1 and given as the input to the second S-box
6. The outputs are XORed with the right side is divided 32bits the output plaintext.
7. The output of step 3 is added to KBS with the last S-bit input and the output is obtaining output of 32bits after combing all the inputs of 8bits by KBS.
8. The outputs of the 4 S-boxes are given to the Mix Column and the output obtained is the final output of the F-function.

3. PROPOSED METHOD FOR MODIFIED ALGORITHM

3.1 Implemented Blowfish Algorithm:

The BLOWFISH Algorithm is the modified name of the Blowfish algorithm that we have enhanced by using the implementation of the older version of Blowfish Algorithm.

The 64 bits of plaintext is given the form of input which is then divided equally into 32bit and is provided to the right side and the left side of the function.

The provided 32bit of left side is EX-OR's with the given key and then further divided into input of 8bit each to the substitution function box namely S-BOX that are of four variants. the output of S-BOX is driven as input by the Key Bits Shifting (KBS).

The output of KBS is of 8bit each which is then combined to become 32bit that's the cipher text which is then providing the output driven to the right side of the plaintext having 32bit. This completes the one round. Same method is used to run and obtain the outputs for the rest 16rounds where the results are swapped using the key functions. After this the outputs are undoing the swap and whitening of the output by the help of given keys that are X-OR'd and helps in obtaining the cipher text.

3.2 F-function

The F-function of BLOWFISH Algorithm consists of 4 8x8 S boxes, and corresponding KBS one 32 bit XOR operation. The total memory consumed for storing the initial hexadecimal digits of the 4 S-boxes are 16 bytes is required to update the values of S-boxes. This increases the overall computational

time and makes it extremely difficult to implement Blowfish on lower end devices. An alternative way to reduce the memory and computational time taken by these S-boxes would be to use the substitution bytes and mix columns concept of modified BLOWFISH algorithm. The S-box is designed such that the correlation between the input and output bits is very low and the mix column helps in mixing the output bytes of the S-box. It uses an 8x8 s-box hence instead of 32-bit XOR gate and 232 modulo adders an 8-bit XOR gate and KBS are used. This also reduces the number of gates required to perform the XOR and modulo addition as compared to the original F-function in Blowfish.

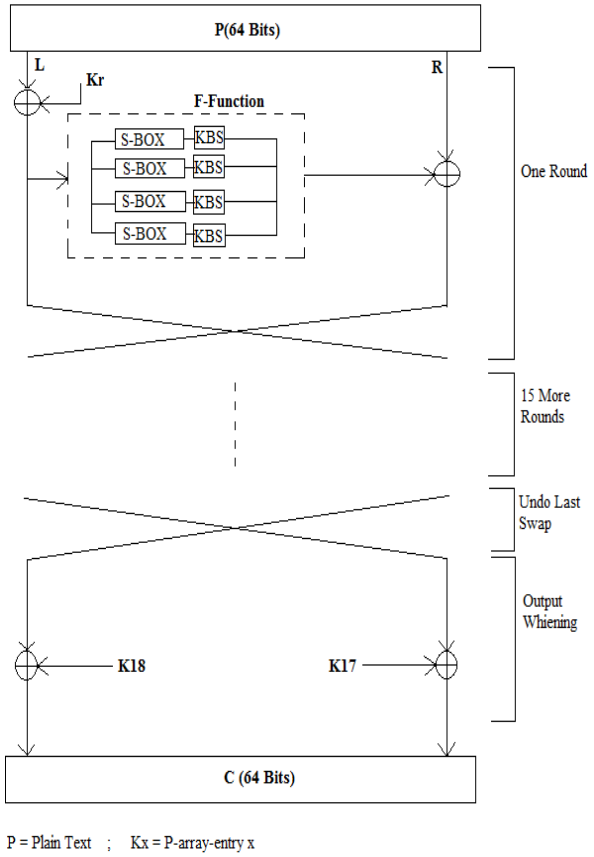

Fig. 3: Implemented Block diagram

3.2.1 Key Bits Shifting (KBS)

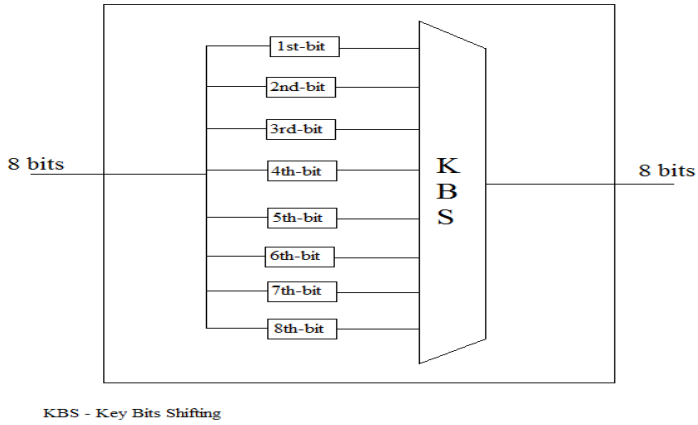

Fig. 4: Key Bits Shifting(KBS)

The 8bits input of S-Box is given to KBS. Where the KBS does the operation of bit shifting of each bits of S-Box output is driven separately. The output of each S-Box is given as input to each 4 KBS. Where the divided input values are shifted using KBS where the output of 4 KBS is combined and output of 32bits obtained. It's X-OR's with right of 32bits and recombines to obtain the 64bits of cipher text.

3.2.3 Advantages Blowfish

- It is somewhat faster than stream cipher as n characters are executed each time.
- Transmission errors in one cipher text block have no effect on other blocks.
- Block ciphers can be easier to implement in software, because they often avoid time-consuming bit manipulations and they operate on data in computer-sized blocks.
- Suitable in trading applications
- In the real-world block ciphers seem to be more general (i.e. they can be used in any of the four modes, the modes is ECB, CBC, OFB, CFB).

3.2.4 Disadvantages Blowfish

- Identical blocks of plaintext produce identical blocks of cipher text.
- Easy to insert or delete blocks and also modify blocks
- Block encryption may be more susceptible to cryptanalysis than stream mode as identical block of plain text yield identical blocks of cipher text.
- Block encryption is more susceptible

4. TOOLS USED

4.1 Xilinx ISE 14.2

ISE Design Suite is an industry-proven solution for All Programmable Xilinx devices. The Xilinx ISE Design Suite continues to bring innovations to a broad base of developers and extends the familiar design flow to all Xilinx FPGA. Engineers can quickly simulate analyze and modify the design without being distracted with implementation details. One can quickly explore different system architectures, evaluating them against key system criteria without investigating effort in writing RTL. This tool was architected to increase the overall productivity for designing with the expanding portfolio of Xilinx devices. These new features and capacity have allowed designers to move a lot more of the overall system design content into the FPGA. Thus the designers are now faced with increased system design integration and verification challenges that require a different design methodology and toolset.

ISE™ controls all the aspects of the design flow. Through the Project Navigator Interface, we can access various design entry and design implementation tools. We can also access the files and documents associated with our project.

4.2 ISim

ISim is a Hardware Description Language (HDL) simulator that lets you perform functional (behavioral) and timing simulations for VHDL, Verilog and mixed-language designs. The ISim environment comprises the following key elements, Vhpcomp (VHDL) and vlogcomp (Verilog) parsers.

- Fuse (HDL elaborators and linkers) command.
- Simulation executable.
- ISim Graphical User Interface (GUI)

5. SIMULATION OUTPUT, RESULTS AND RTL SCHEMATIC

5.1 Results

The strength of any cipher can be measured through various tests and standard criterions used in the field of cryptography. Before going into the strength test results, we shall start by discussing few standard criterions used to measure the strength of ciphers.

- **Shannon's Confusion and Diffusion property:** This property was proposed by Claude Shannon and is used to

determine whether a cipher is secure or not. The Confusion property states that the relation between the plaintext and the ciphertext must be highly non-linear, The **Diffusion** property states that a small change in the plaintext or key must affect a large change in the ciphertext.

- **Avalanche effect:** It refers to a desirable effect similar to that of diffusion wherein a small change in the input bits produces a significant change in the output bits Strict Avalanche Criterion: This is an extension of the diffusion property. It states that in a secure cipher whenever a single input bit is changed at least half of the output bits must change.
- **Bit Independence Criterion:** It states that whenever an input bit i is changed the output bits j and k must change independently for all i, j and k.
- **Number of rounds:** The greater the number of rounds the more difficult it is to break the code even for a weak F-function. Blowfish uses Feistel network which consists of 16 rounds has proven to resist cryptanalysis.
- **F-function:** The confusion element for the cipher is provided by the F-function which produces an output that is completely different compared to the input. The F-function must be highly complex to ensure that there is maximum confusion between the input and the output.
- **Keyalgorithm:** It is found in the case of simplified DES and other weak ciphers the algorithm was easily breakable because the key used was very short. In general, longer the key better is the security. Blowfish uses a 448-bit key and is found to be secure enough.

5.2 RTL Schematic

After looking into the security aspects of the modified algorithm we look next like to concentrate on the design aspects. The RTL schematic view of the modified F-function Key extension and 128-bits extension is shown. The RTL schematic gives a general view regarding the logic gates used and interconnection between various components of the algorithm.

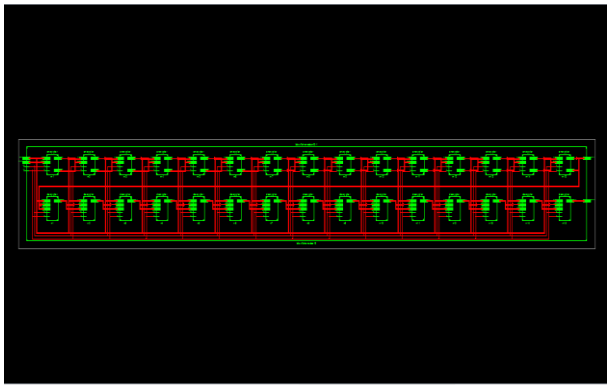

Fig. 5 : RTL Over View

5.3 Simulation Output

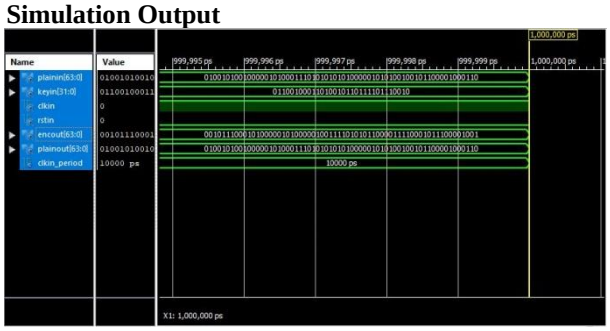

Fig. 6: Output in Binary

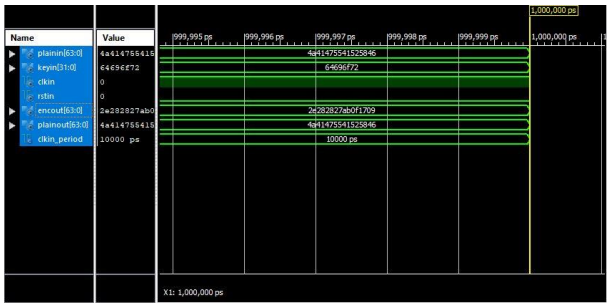

Fig. 7: Output in Hexadecimal

6. CONCLUSION

The principle and working of the Blowfish Algorithm were briefly discussed in this paper. It was found from the analysis of blowfish algorithm that the quality of the cipher text can be improved by adding the concepts of S-BOX and KBS. The results obtained were quite appreciative and the complexity of the key and the cipher text were also found to be good. This concept is present simulated and synthesized as software and the concept has been implemented to make it as a unique one for the sender and receiver. Blowfish Algorithm is made by implementing the modified blowfish algorithm by making that cryptography more enhanced and secured.

4. REFERENCES

[1] Tingyuan Nie, and Teng Zhang, “A Study of DES and Blowfish Encryption) Algorithm,” 978-1-4244-45479/09/IEEE, TENCON 2009.

[2] Brian Cody, Justin Madigan, Spencer MacDonald, Kenneth W. Hsu, “High Speed SOC Design for Blowfish Cryptographic Algorithm,” 2007 IFIP International Conference on Very Large Scale Integration (VLSI-SoC 2007), 978-1-4244-1710-0/07.

[3] B. Schneier the Blowfish Encryption Algorithm.http://www.schneier.com/blowfish.html

[4] Raja Jitendra Nayaka, and Jamuna.S, “FPGA Implimentation of modified Blowfish Algorithm” Dayanand Sagar college of Engineering 2012-2013

[5] D. Manoj Kumar, Dr. M. Sundhararajan “VHDL Implementation Using Modified Blowfish Algorithm”, February 2015

[6] William Stallings, “Cryptography and Network Security”, 5th edition, Pearson Publications.

[7] B. Schneier. The Blowfish Encryption Algorithm. http://www.schneier.com/blowfish.html

[8] Modified Blowfish Algorithm, October 2018

[9] Arya S “An Implementation of Blowfish Algorithm Using FPGA”, August – 2013

[10] S. Sweetlin Susilabai, D.S. Mahendran, S. John Peter “Interbit Exchange and Merge (IBEM) Pattern of Blowfish Algorithm”, January 2019

[11] Tingyuan Nie, and Teng Zhang, “A Study of DES and Blowfish Encryption) Algorithm,” 978-1-4244-4547-9/09/IEEE, TENCON 2009.

[12] William Stallings, “Cryptography and Network Security”, 5th edition, Pearson Publications.

