International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 8, August - 2013

An Implementation of Blowfish Algorithm Using FPGA

Arya S

Dept. of ECE
Sree Buddha College of Engineering, Alappuzha
Kerala ,India

Abstract

Blowfish is a symmetric key cryptographic algorithm. It is a Feistel network, iterating a simple encryption function 16 times. The block size is 64 bits, and the key can be any length up to 448 bits. . It is very fast and useful scheme for encryption and decryption. Blowfish is a keyed, symmetric block cipher, designed in 1993 by Bruce Schneier and included in a large number of cipher suites and encryption products. It is designed to meet the goals such as speed, compactness, simplicity etc. The Blowfish algorithm consists of two steps including key expansion and data encryption. Key expansion is used to generate the subkeys for encryption, using the subkeys the data is encrypted and finally the encrypted data is decrypted. The decryption is exactly same as that of the encryption, except that the keys are used in the reverse order. Here the encryption with blowfish algorithm can be done using FPGA platform and code can be written

1. Introduction

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by any-one except the intended recipient. While cryptography is the science of securing data, cryptanalysis is the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck.

Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system but it is infeasible to do so by any known practical means. These schemes are therefore termed computationally secure; theoretical advances (e.g., improvements in integer factorization algorithms) and faster computing technology require these solutions to be continually adapted. Symmetric algorithms, sometimes called conventional algorithms, are algorithms where the encryption key can be calculated from the decryption key and vice versa. In most symmetric algorithms, the encryption key and the decryption key are the same. These algorithms, also called secret-key algorithms, single-key algorithms, or one-key algorithms, require that the sender and receiver agree on a key before they can communicate securely. The security of a symmetric algorithm rests in the key; divulging the key means that anyone could encrypt and decrypt messages. As long as the communication needs to remain secret, the key must remain secret. Blowfish is a symmetric cryptographic algorithm.

2. Blowfish Algorithm

In 1993, Bruce Schneier published the Blowfish block cipher. At this time the current Data Encryption Standard (DES) was known to be vulnerable to cryptanalysis and brute-force attacks. Other cryptographic algorithms were available to replace DES, but many of the cryptographic algorithms were either protected by patents or considered proprietary. Schneier developed Blowfish to be a publicly available cryptographic algorithm with the potential to replace DES. Schneier also encouraged others to evaluate the performance and security of Blowfish. Blowfish is a variable-length key block cipher. It does not meet all the requirements for a new cryptographic standard discussed above: it is only suitable for applications where the key does not change often, like a communications link or an automatic file encryptor. It is significantly faster than DES when implemented on 32-bit microprocessors with large data caches, such as the Pentium and the PowerPC. The message without encryption is the plain text and encrypted message is called cipher text. The algorithm involves two steps such as key expansion and data encryption.

2.1 Subkeys

Blowfish uses a large number of subkeys. These keys must be pre computed before any data encryption or decryption.

1. The P-array consists of 18 32-bit subkeys:
$P_1, P_{2\ldots}, P_{18}$.
2. There are four 32-bit S-boxes with 256 entries each:
$S_{1,0}, S_{1,1}, ., S_{1,255}$;
$S_{2,0}, S_{2,1}, .., S_{2,255}$;
$S_{3,0}, S_{3,1}, ..., S_{3,255}$;
$S_{4,0}, S_{4,1}, .., S_{4,255}$;
The exact method used to calculate these subkeys will be described later.

2.2 Data Encryption

Blowfish is a Feistel network consisting of 16 rounds (Figure.1). The input is a 64-bit data element, X. Divide x into two 32-bit halves: X_L, X_R

- For i = 1 to 16:

$$X_L = X_L \text{ XOR } P_i \quad (1)$$

$$X_R = F(X_L) \text{ XOR } X_R \quad (2)$$

- Swap X_L and X_R

$$X_R = X_R \text{ XOR } P_{17} \quad (3)$$

$$X_L = X_L \text{ XOR } P_{18} \quad (4)$$

- Recombine X_L and X_R

Function F is given as;

Divide X_L into four eight-bit quarters: a, b, c, and d

$$F(X_L) = ((S1,a + S2,b \bmod 2^{32}) \text{ XOR } S3,c) + S4,d \bmod 2^{32} \quad (5)$$

Decryption is exactly the same as encryption, except that $P_1, P_2, ..., P_{18}$ are used in the reverse order.

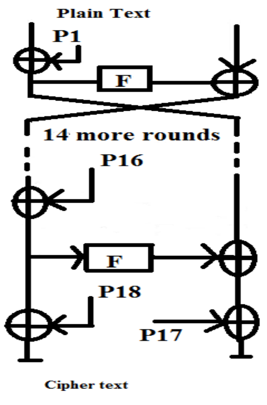

Figure1. Feistel Network

3. Generation of Subkeys

The exact method for calculating the subkeys involves the following steps:

- Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the hexadecimal digits of pi like the following example:
  P1 = 0x243f6a88
  P2 = 0x85a308d3
  P3 = 0x13198a2e
  P4 = 0x03707344

The block diagram for F function is shown in Figure 2.

- XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key .Repeat the cycle through the key bits until the entire P-array has been XORed with key bits.
- Encrypt the all-zero string with the Blowfish algorithm, using the subkeys described in steps (1) and (2).
- Replace P1 and P2 with the output of step (3).
- Encrypt the output of step (3) using the Blowfish algorithm with the modified sub keys.
- Replace P3 and P4 with the output of step (5).
- Continue the process until the entire P values have been replaced.

There is 521 iteration needed to generate these processes.

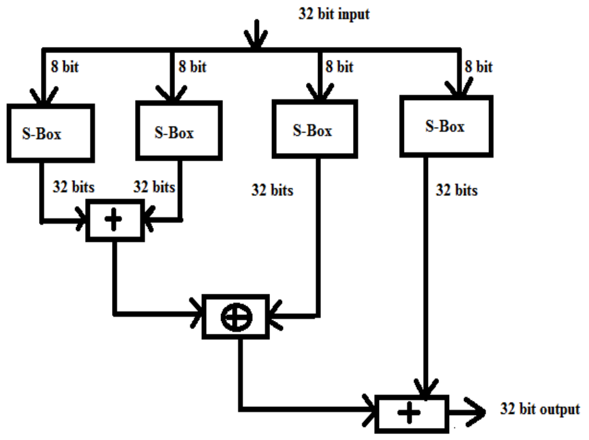

Figure2. Generation of F function.

4. Results and Discussions

The code were written in VHDL and simulated with the VHDL simulator. The VHDL simulator ISIM is used here for the simulation. The key generation involving various steps and that can be used to generate the actual key for encryption. The decryption is done exactly same as the encryption except that the keys are used in the reverse order.

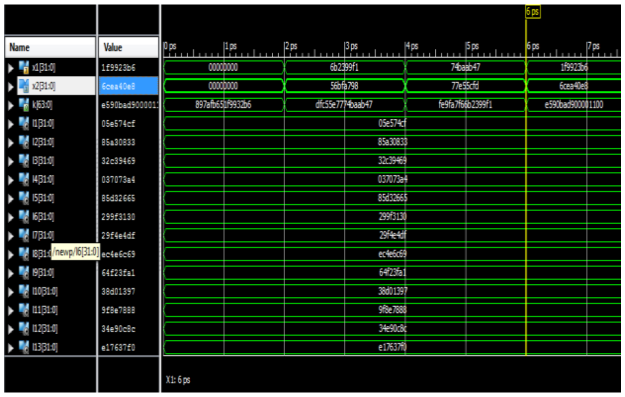

Figure 3. Key Generation.

The subkeys are generated before encryption. Due to the complicating steps for the generation of the subkeys helps the algorithm to prevent from cryptanalysis. Using the subkeys generated the data is encrypted and decryption is the reverse process of encryption.

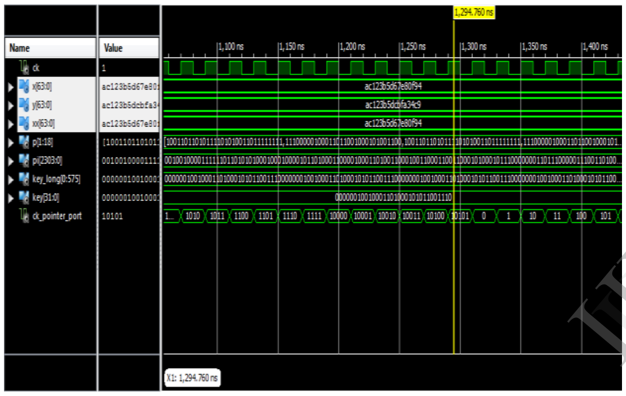

Figure 4.Simulated result of encryption and decryption.

The encryption and decrypted result is shown in Figure 4, it is clear that the data is encrypted and after decrypting with the same key the original data is obtained. The x shows the data to be encrypted, y shows the encrypted data and xx shows the decrypted data, it is same as the input data.

5. Conclusions

The Blowfish algorithm is designed and coded in VHDL and simulated using VHDL simulator and the simulated results are obtained and verified. The hardware platform selected to implement the algorithm is FPGA. Blowfish is fastest, compact, simple, and secure than other encryption algorithm, and implementing in FPGA results in simplifying the complexity in coding, less cost, flexible, reduced time consuming, power reduction etc. At first the secret key is generated and using the key generated the data is encrypted. The decryption is also done using the VHDL for verification and it is same as that of encryption and the only difference is the key is used in the reverse order. The flexibility of the key makes blowfish more secure than other existing algorithms. The key generation is the difficult task; due to the variable key length it is not easy to break the algorithm. Hardware implementation of the high secured algorithm is used for high speed applications and also the plain text is encrypted in a small amount of time.

6. References

[1]Russel K.Meyers., Ahmed H.Desoky,"An Implementation of the Blowfish Cryptosystem", *IEEE International Symposium on Signal Processing and Information Technology* (ISSPIT'08), 2008, pp. 346-351.

[2]Michael C.-J. Lin,Youn-Long Lin," A VLSI Implementation of Blowfish Encryption/Decryption Algorithm", *IEEE Design Automation Conference* (A" SP_DAC'00), 2000, pp.1-2.

[3] Whitfield Diffie and Martin Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, vol. IT-22, pp: 644–654, Nov. 1976.

[4]Brian Cody, Justin Madigan," High Speed SOC Design for Blowfish Cryptographic Algorithm", IEEE *International Conference on VLSI* (IFIP'07), 2007, pp.284-287.

[5]Bruce Schneier,"*Applied Cryptography*", 2nd Edition, Chapter 13-9,pp.39-353,Wiley editions,1996.

[6] Monika Agrawal, Pradeep Mishra," A Modified Approach for Symmetric Key Cryptography Based on Blowfish Algorithm", *International Journal of Engineering and Advanced Technology* (IJEAT), vol.1,no.6,pp. 2248-2255, Aug 2012.

[7]Allam Mousa, "Data Encryption Performance Based on Blowfish", 47th International Symposium (ELMAR-2005),pp.08-1 0, June 2005.

[8]William stallings,*" Cryptography and Network Security, Principles and Practices",* 2nd Edition, Chapter 3-6, pp.106-107,197-202, Pearson Education 2003.

[9]Atul Kahate, "*Cryptography and Network Security*", 5th Reprint, Chapter 3-4, pp.63-159,Tata McGraw-Hill,New Delhi, 2005.