Nile Journal for Commercial, Legal and Information Systems Science
Volume 1, Number 1, April

مجلة النيل للعلوم التجارية والقانونية و نظم

ARTICLE INFO.

Article History
Received: 8 January 2021
Received in Revised form: 14 January 2021
Accepted: 2 February 2021

Keywords
Management
Accountant

Blowfish cryptography implementation by using Microcontroller

Ali Takieldeen[1], and Heba Soltan[2]

[1] Air Defence College, (dr.ali.takieldeen.5350.adc@alexu.edu.eg)
[2] lecturer at Nile higher Institute for Commercial Science
and Computer Technology in Mansoura
Correspondence: a_takieldeen@yahoo.com

Abstract:

The main task of paper studied new ciphering and deciphering techniques reported previously. Computer programs are designed in C language to perform theses algorithms and to assess their performance. Out of these algorithms, it is found that the BLOWFISH is the more sophisticated one. Therefore, a development for this algorithm is introduced to enhance its operation. This has been carried out by using the Microcontroller in the decrypting operation to support fast and more accurate operation. From the results of simulation, it is found that the expansion and permutation operations require most of the computing time. Therefore, software was designed and implemented to execute the expansion and permutation operations. A series of experiments was connected using the new developed algorithm as applied to different types of data (e.g. text, graphics and geographic maps). In all paper experiment, the encrypted and decrypted files were exactly the same.

Keywords

Blowfish Algorithm; AVR Microcontroller

1. Blowfish Algorithm:

The Blowfish algorithm uses a key [1], which can be up to 448 bits in length, to calculate sub-keys, which are used in the actual encryption and decryption. The sub keys used in Blowfish are the P-array and S-boxes.

The P-array consists of 18 32-bit values while the 4 S-boxes consist of 256 32-bit values. The original values of the P-array and the 4 S-boxes are initialized with the hexadecimal fraction of pi. The P-array is initialized first and then followed by the S-boxes. The hexadecimal fraction of pi was selected because it produced a random number for the initialization. Another type of initialization could be used but the initialization values must be random. According to Schneier, patterns in the initialization values can result in a weaker cipher. After initialization, the values of the P-array and S-boxes are modified using the key. The key [2], which can consist up to

448-bits, is segmented into 32-bit values. If the key is less than 448-bits, the key is repeated. For a 448-bit key, there would be 14 32-bit values, which mean the key-array (K) has 4 32-bit values less than the P-array.

The key is then used to initialize the P-array. An exclusive-or operation is performed between each of the 18 32-bit P-array values and a 32-bit value of the key. P1 is XOR-ed with the first 32-bits of the key (K1) and then P2 is XOR-ed with the next 32- bits of the key (K2). This process is continued until P15 because the key-array has only 14 32-bits. According to Stallings, an exclusive-or operation would be conducted with P15 and the first 32-bit key value (K1), which is incorrect. The key value is actually the next value of the key repetition, which would essentially be K15. After the P-array has been initialized with the key, the P-array and S-boxes values are then modified using the encryption routine of Blowfish. First, two 32-bits values consisting of zeros is encrypted. Other initial values could be selected but these values must be fixed to prevent an attacker from generating the same cipher text with two different keys. The ciphertext [3] from the encryption of the zero values is used to replace the P-array values P1 and P2.

The cipher text is also used as the input for the next encryption round. The cipher text from this round then replaces P3 and P4 and the encryption routine is executed again with the cipher text as input. This process [4] is continued until all 18 values in the P-array and all 256 values for each of the 4 S-boxes is replaced with cipher text. A total of 521 encryptions are preformed to obtain all the P-array and S-box values. Now that the P-array and S-box values have been established, plaintext can now be encrypted. For encryption [5], the 64-bit plaintext is separated into a left and right half each consisting of 32-bits. The encryption routine consists of a 16 round Feistel network. In the first round, an exclusive-or operation is performed between the left 32-bits (LE-0) and the 32-bit P1 of the P-array. This value becomes the next 32-bit right value (RE-1) and this value is also inserted into the F function.

The encryption routine for Blowfish uses a 16 round Feistel network, a swap operation and two exclusive-or operations. Each round consists of exclusive-or operations and the F function. The F function [6] takes the 32-bit input and separates it into 4 bytes (8-bits each). These four values are then used for table lookup in their respective S-Boxes. A 32-bit input is parsed into 4 8-bit values that are used for the table lookup into the 4 S-boxes. The 32- bit values of the S-boxes are then added, exclusive-or-ed and then added again. Unlike DES [7] which uses more bits to map to S-boxes, Blowfish only uses four 8-bit values for mapping to the S-box values. A less complex mapping was used because the S-boxes are generated from the key values and are not static like DES. The 32-bit values of the S-boxes are then manipulated according to the following formula:

$$32\text{-bit Output}= ((((S[1][a] +S[2][b]) \bmod 2^{\wedge}32) \oplus S[3][c])+S[4][d]) \bmod 2^{\wedge}32 \qquad (1)$$

The 32-bit value of S-box 1 is added to the 32-bit value of S-box 2. The modulus of this result by 2^32 is taken as the input for the exclusive-or operation to performed with the 32-bit value of S-box 3. The result of the exclusive-or operation is then added to the 32- bit value from S-box 4 and the modulus 2^32 is then performed. A bitwise exclusive-or operation is performed on the final 32-bit output from the F function and the right half of the data (RE-0). The result of this operation becomes the left half 32-bit input for the next round (LE-1). The results of round 1 can be explained by the following equations:

$$LE\text{-}1 = F(LE\text{-}0 \oplus P1) \oplus RE\text{-}0 \qquad (2)$$

$$RE\text{-}1 = LE\text{-}0 \oplus P1 \qquad (3)$$

Round 2 is then performed with inputs LE-1 and RE-1. This process is repeated for a total of 16 rounds. The general equations to describe the rounds are as follows:

$$LEi = F(LEi\text{-}1 \oplus Pi) \oplus REi\text{-}1 \text{ where } 1 \leq i \leq 16 \qquad (4)$$

$$REi = LEi\text{-}1 \oplus Pi \text{ where } 1 \leq i \leq 16 \qquad (5)$$

After completing the 16 rounds, (LE-16) and (RE-16) values are swapped. An exclusive-or operation is then performed between the swapped (LE-16) and P18 and also with the swapped RE-16 and P17 to obtain (LE-17) and (RE-17), respectively. The 32-bit values of (LE-17) and (RE-16) are combined to obtain the 64-bit cipher text. The decryption process for Blowfish [8] is almost identical to the encryption process except the P-array values are reversed. For decryption, the bitwise exclusive or operation is performed between the first left 32-bit value (LE-0) of the cipher text and P18. In the encryption, this process would have been performed with P1. The decryption process is repeated for the 16 rounds. (LE-16) and (RE-16) are then swapped and a bitwise exclusive-or operation is performed with P1 and (LE-16) and also with P2 and RE‘-16 to obtain (LE-17) and (RE-17), respectively. (LE-17) and (RE-17) are then combined to obtain the original plaintext. The decryption routine for Blowfish is identical to the encryption routine except the P-array key values are applied in the reverse order.

2. AVR Microcontroller:

Microcontroller is microcomputer [9] that contains most of its peripherals and required memory inside a single integrated circuit along with the CPU (microcomputer on a chip microcontroller have been in use for more than three decades. Intel 8051 was one of the first microcontrollers in the market Other companies manufacture microcontrollers such as National, Motorola,Philips,Zilog , Hitachi, Microchip and Atmel. Atmel has become a world leader in the development of FLASH memory technology. FLASH technology [10] is a nonvolatile. Yet reprogrammable memory often used in products such as digital cameras and portable audio devices. This memory technology really pushed Atmel ahead in the microcontroller industry by providing in system programmable solution. The next great step in this high tech revolution was the Implementation of high level language compilers that are targeted specifically for use with new microcontrollers. The code generation and optimization of the Compilers is quite impressive. The C language lends itself can be considered the Best choice since it creates pools intellectual property that can be drawn from Again and again. This lowers the development costs on an ongoing basis by shortening the development Cycle with each subsequent design.

Why AVR:

a. World's Best Flash MCU!
b. High CPU Performance (One Instruction Per Clock Cycle).
c. High System Integration.
d. Small Code Size.
e. Onboard Hardware Multiplier.
f. 16 Bit Performance at 8 Bit Data bus.
g. Operates on 5 volt DC.
h. Can be programmed in seconds with single power supply via simple 4 pin connector
i. Many upgrade Options (AVR , MEGA AVR , LCD AVR , TINY AVR ,USB AVR , RF AVR , SECURE AVR)
j. AVR core use it in FPGA and FPGA works as microprocessor

3. Blowfish cryptography implementation by using Microcontroller:

The Blowfish algorithm uses a key, which can be up to 448 bits in length, to calculate sub-keys, which are used in the actual encryption and decryption. The sub keys used in Blowfish are the P-array and S-boxes. The P-array consists of 18 32-bit values while the 4 S-boxes consist of 4 * 256 32-bit values. The original values of the P-array and the 4 S-boxes are initialized with the Hexadecimal fraction of pi. The P-array is initialized first and then followed by the S-boxes. The hexadecimal fraction of pi was selected because it produced a random number for the initialization. Another type of initialization could be used but the initialization 5 values must be random.

3.1 Encryption:

In the beginning we encrypt the file text using PC Software that we developed to read the 1st 46 bit from the plaintext file then, separate into two (left and right) parts each is 32bit. The encryption operations consist of 18 rounds, the 1st 16 rounds are similar but the last two are a swap operation.

In the 1st round, the left 32bit are XORed with the 32bit key and the o/p of this operation exchange the 32bit according to the lookup table.

The O/P of the lookup table is 32bit XORed with the Right 32bit. That round is looped up to 16 times, and then we get the O/P as the encrypted file and put it on the MMC which is connected to the PC.

3.2 Decryption:

The microcontroller on the receiver reads the ciphered text from the MMC by a serial protocol called Serial peripheral interface (SPI). The microcontroller starts the 1st cycle of decrypting the plain text while reading the 2nd 64bit at the same time (two cycles); at the same time it reads the key from the O/P data port of the LCD [11]. The decryption operation on the microcontroller is the same as it on the PC software except that it's in a reverse order of key (starting with key 18) . The microcontroller sends the decrypted file from the O/P port to the data I/p port of the graphical LCD, and then we can get the original plaintext displayed on the screen.

4. CONCLUSIONS:

This paper, designed an original Blowfish encryption algorithm in hardware implementation, and a proposed Blowfish Block Cipher for both hardware and software implementation. Both of them are a symmetric key block cipher with a 64-bit input/output block, and key length is up to 448 bits for Blowfish, and 128 bits input key for the proposed.

The proposed Blowfish algorithm using microcontroller is a fast and secure cipher that is characterized by its dynamic, which adds much strength to it. Also, this dynamic nature enforces exhaustive-search over the key space. The present algorithm has a powerful carefully designed key scheduling. And can enable us to overcome the higher defect that is appearing in the blowfish algorithm.

Advantages:

a. Easy to implement in both hardware and software.
b. Relatively inexpensive to implement in hardware.
c. It‘s a robust block cipher algorithm.
d. It's a high performance algorithm.

The proposed design helps in accelerating the computation of the algorithm comparing with original Blowfish algorithm and other previous implementations.

REFERENCES:

[1] N. Ferguson and B. Schneier, *Practical Cryptography*, Wiley, 2003.

[2] Nikooghadam, M.; Amintoosi, H. Cryptanalysis of Khatoon et al.’s ECC-based Authentication Protocol for Healthcare Systems. arXiv preprint arXiv:1906.08424 2019.

[3] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, J.Wiley, 1996.

[4] Khatoon, S.; Rahman, S.M.M.; Alrubaian, M.; Alamri, A. Privacy-Preserved, Provable Secure, Mutually Authenticated Key Agreement Protocol for Healthcare in a Smart City Environment. IEEE Access 2019, 7, 47962–47971.

[5] El-Sayed A. El-Badway, Abdel-Aty M. Emarah, and Ali E. Taki El-Deen. "Performance Improvement of Data Encryption Standard," The 7th world Multi-Conference on Systemics, Cybernteics and Informatics (SCI 2003), Orlando, Florida, USA, July 2003.

[6] R. Schneier, "The Blowfish Encryption Algorithm," Dr. Dobb's Journal, April 1994.

[7] Malik, M.; Dutta, M.; Granjal, J. A survey of key bootstrapping protocols based on public key cryptography in the Internet of Things. IEEE Access 2019, 7, 27443–27464.

[8] Froomkin, A.M.,: "The Metaphor is the Key: Cryptography, The Clipper Chip, And the Constitution", Microsoft Internet Explorer, May 16, 1995.

[9] AVR using code vision c compiler First Edition Author: Mohamed Sobky.

[10] Multimedia Card™ HITACHI data sheet.

[11] SED 1335 LCD Controller ICs data sheet.