Discrete Mathematics
Final Project

Poojan Shah - 2021 01132
Dhruvil Thakor - 202101462
Atharva Chaudhari - 202101460
Izhan Sheth - 202101124
Aum Patel - 202101118

July 2022

1 Introduction

The Main objective of the project is to give a background about the encryption and decryption of data using the Blowfish Algorithm. Developed by Bruce Schneier in 1993. It is a symmetric-key block cipher used in data encryption of passwords, emails and bank account details.

The paper is divided into the following subdivisions:

a) Motivation for the project
b) What is a Feistel Cipher?
c) Working of the Blowfish Algorithm
d) Advantages of the Algorithm
e) Disadvantages of the Algorithm
f) Significance of the Blowfish Algorithm
g) Bibliography

2 Motivation for the project

Problem statement:

The problem statement of the project is based on how the data is being sent across the internet in a secured manner along with the system being used.

3 What is a Feistel Cipher?

The Feistel Cipher is a cipher that works by splitting large data blocks into 2 equal blocks. The encryption algorithm is then applied to these blocks after which it is run in a loop. The number of rounds varies for different ciphers that use Feistel network. The network will also give the original input if the output is run through the loop in reverse.

The Blowfish Algorithm was developed as an alternative to DES (Data Encryption Standard). It is significantly faster than DES and provides a good encryption rate with no effective cryptanalysis technique found to date. It is a variable length, symmetric, 64-bit block cipher featuring a 64-bit block size and taking a variable length key from 32bits to 448 bits. It uses 16 Feistel-like iterations where the iteration operation takes place on a 64-bit block, split into two 32 -bit blocks.

4 Working of the Blowfish Algorithm

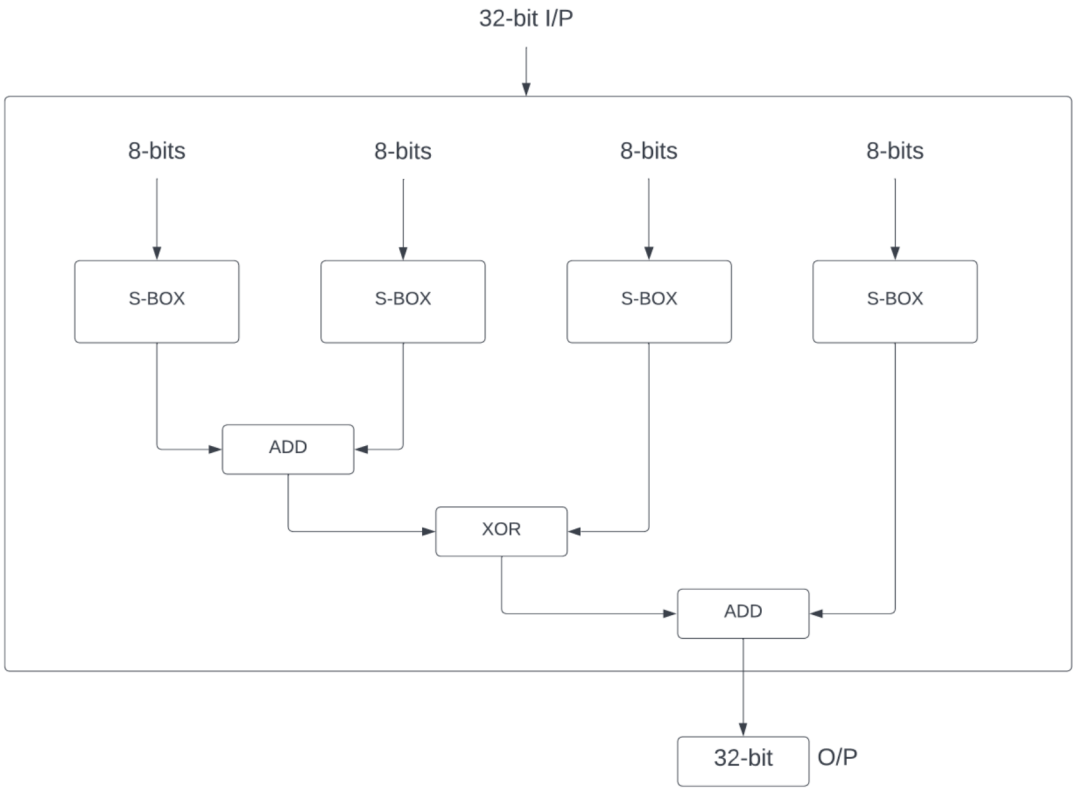
Blowfish makes use of a 16-round Feistel network for encryption. Each round consists of a key-dependent permutation and a key and data dependent substitution. All the operations that take place use XOR gate – a logic gate and summation operations.

Let's look at some terms before we get into how the algorithm works.

- Block size: 64 bits
- Key size: Variable from 32 bits to 448 bits
- Number of sub keys: 18 (P-array)
- Number of rounds: 16
- Number of substitution boxes(S-boxes): 4 (each have 512 entries of 32-bits)

Function F: This is a function used in the algorithm where a 32-bits input is split into four 8-bit inputs. These 8-bit inputs are then run through a S box which changes them back to a 32-bit output. The summation and XOR operation are then performed on these outputs till we get a final 32-bit output.

The algorithm uses P-array and S-boxes. P-arrays consists of eighteen 32-bit subkeys and S-boxes has 256 entries. The values for these are to be initialized by the user but usually they are initialized with a fixed string of hexadecimal digits of pi.
The image below demonstrates this:


32-bit I/P
8-bits
8-bits
8-bits
8-bits
S-BOX
S-BOX
S-BOX
S-BOX
ADD
XOR
ADD
32-bit
O/P

Generation of Sub keys:

First, the P-array is to be initialized with values. Here hexadecimals digits of pi are used.

Ex : o P1=0x243f6a88,
o P2=0x85a308d3,
o P3=0x13198a2e,
o P4=0x3707344, etc.

1- p1 is XORed with the first 32 bits of the key

2 - The output of this is then run through a function F

3- the output of this function is then XORed with second half of the key

4- at last the output of the first XOR operation is swapped with the output of the last XOR operation.

5- This is performed for a total of 16 times.

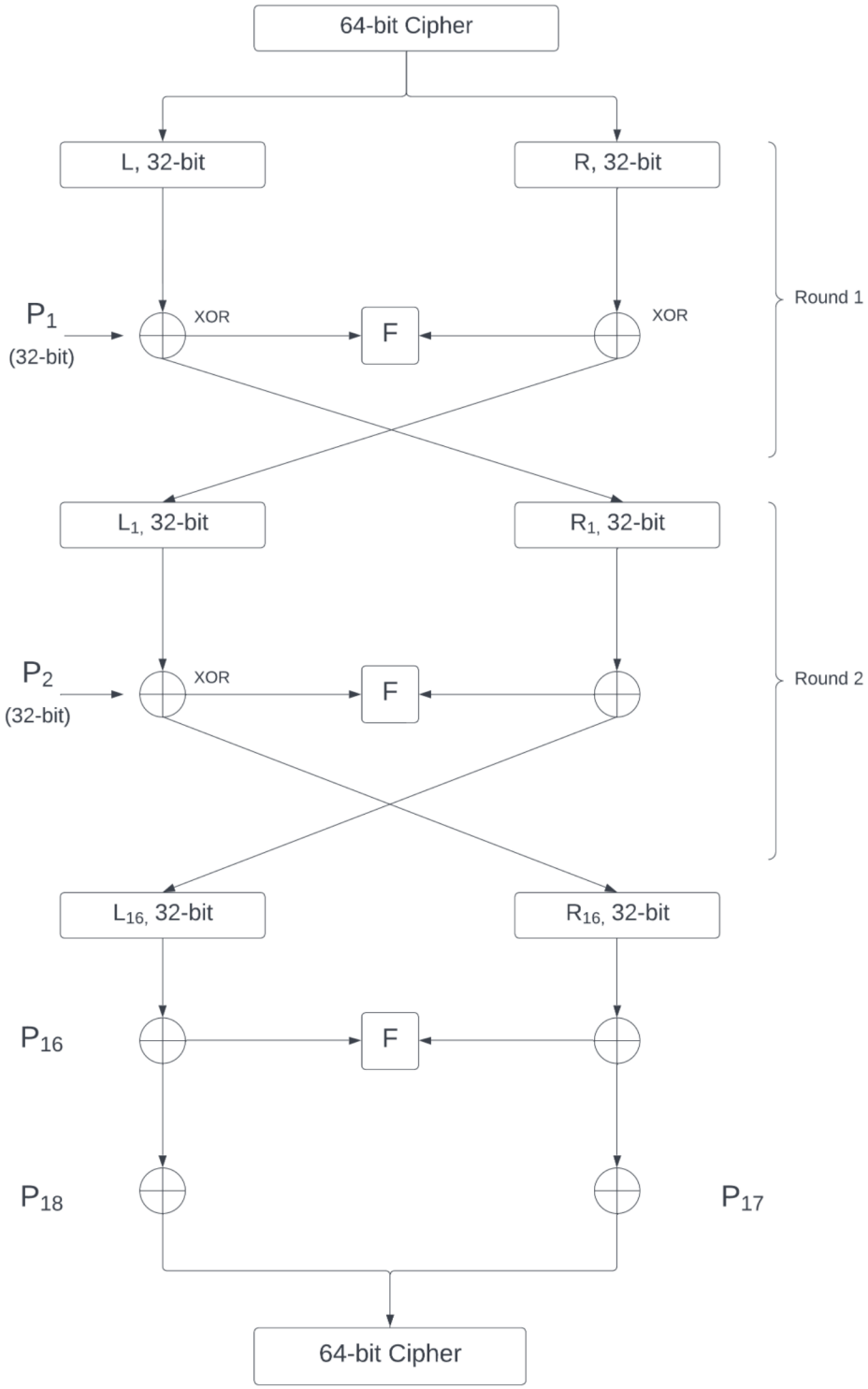
6- A final swap is performed at the end and the Left and Right 32-bit texts are concatenated to get the final cipher text.

Input is a 64-bit data element: X
X is divided into two 32-bit halves: xL and xR

• For i = 1 to 16
o xL = xL XOR pi
o xR = F (xL) XOR xR
o Swap xL, xR

• Undo last swap
o xR = xR XOR pi (i=17)
o xL = xL XOR pi (i=18)

• Concatenate xL and xR to get 64-bit cipher text
To get back the original Input all we have to do is run the same algorithm in the reverse order with the ciphered text as the input.


64-bit Cipher
L, 32-bit
R, 32-bit
Round 1
P1
(32-bit)
XOR
F
XOR
L1, 32-bit
R1, 32-bit
P2
(32-bit)
XOR
F
Round 2
L16, 32-bit
R16, 32-bit
P16
F
P18
P17
64-bit Cipher

5 Advantages of the Blowfish Algorithm

1. Faster than other encryption algorithms such as the DES.

2. The algorithm is unpatented and thus is free to use by anyone.

3. Brute force attacks are difficult as key expansion of Blowfish takes a long time.

6 Disadvantages of the Blowfish Algorithm

1. The Algorithm only allows 64-bit inputs which is limiting.

2. Key expansion for the algorithm takes a long time.

3. The Blowfish algorithm can't provide authentication as two people can have the same key.

4. Two fish algorithm that was developed after blowfish solves many of these problems.

7 Significance of the Blowfish algorithm

Blowfish is an encryption technique that is a highly powerful tool against hackers and cybercriminals, despite the fact that you could think of it as simply a cute aquarium fish. It is included into many different devices, such as TiVo, backup software, safe email encryption tools, and systems for managing passwords

References

[1] Chaitali Haldankar and Sonia Kuwelkar. Implementation of aes and blowfish algorithm. *International Journal of Research in Engineering and Technology*, 3(03):143–146, 2014.

[2] Bruce Schneier. *Schneier on security*. John Wiley & Sons, 2009.
[2] [1]