Thuật toán Blowfish

Họ tên: Hồ Thành Nguyên
MSSV: 10344991
Lớp: DHTH6ALT

1. **Lịch sử của thuật toán Blowfish:**

Blowfish được phát triển bởi Bruce Schneier. Lý do thuật toán này được phát triển là do:

- DES đã bị đi vào giai đoạn cuối và cần được thay thế.
- Một số thuật toán thay thế khác được bảo hộ bởi bằng sáng chế.

Blowfish dùng mạng Feistel để mã hóa. Blowfish là một thuật toán mã hóa khối 64 bit. Có thể mã hóa bất cứ chiều dài nào lên tới 448 bit.

2. **Khái quát thuật toán Blowfish:**

Blowfish dùng 18 khóa con và 4 khóa phụ thuộc s-boxes. Những khóa và s-boxes này được tính toán trước khi mã hóa thành công. Sự mở rộng của khóa này được giải thích ở mục 1.5. Sơ đồ mã hóa Blowfish được thể hiện ở Hình 1. Sơ đồ mã hóa Blowfish dùng phép toán làm tròn cho 16 lần lặp. Vào vòng lặp cuối cùng, giá trị trái và phải được hoán vị và phép toán XOR cho khóa con P17 và P18 . Chú ý rằng phép làm tròn này khác hơn so với DES. DES không hoán vị nửa giá trị trái và phải trong vòng lặp cuối này.

3. **Phép làm tròn Festel (Blowfish):**

Vòng lặp Blowfish là một thuật toán 16 lần đơn giản. Phép làm tròn Feistel (Blowfish) được thể hiện ở hình 2 và phép toán làm tròn (F) được thể hiện ở hình 3. Giá trị bên trái là kết quả của phép XOR với khóa con. Kết quả này được hoán vị với giá trị bên phải. kết quả cũng được dùng để tính toán phép làm tròn dùng XOR với phần dữ liệu bên phải. Có thể được viết như sau:

$$\mathbf{Li = F(Li{-}1 \oplus Pi) \oplus Ri{-}1}$$

Và

$$\mathbf{Ri = Li{-}1 \oplus Pi}$$

4. **Phép toán làm tròn Blowfish:**

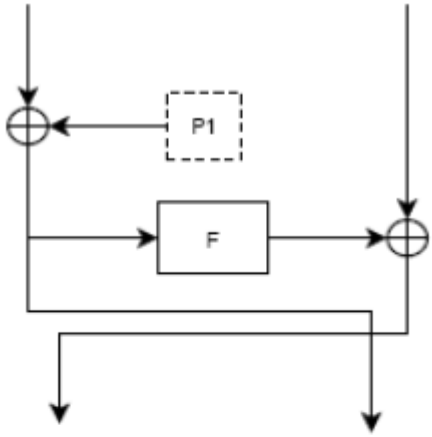
Để làm tròn, dữ liệu 32 bit được chia thành 4 dãy số 8 bit. Sau đó, phép làm tròn sẽ thực hiện phép tính như sau:

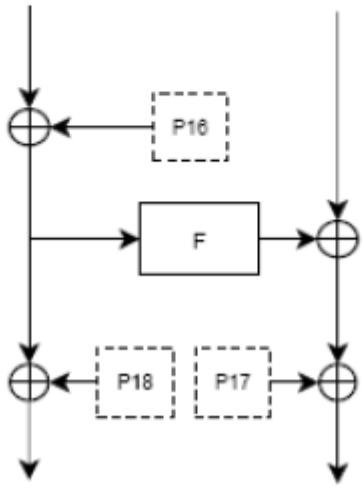
$$\mathbf{F(Xl) = (((S1,a + S2,b) \bmod 232) \oplus S3,c + S4,d) \bmod 232}$$

Trong đó, "+" là phép cộng số nguyên

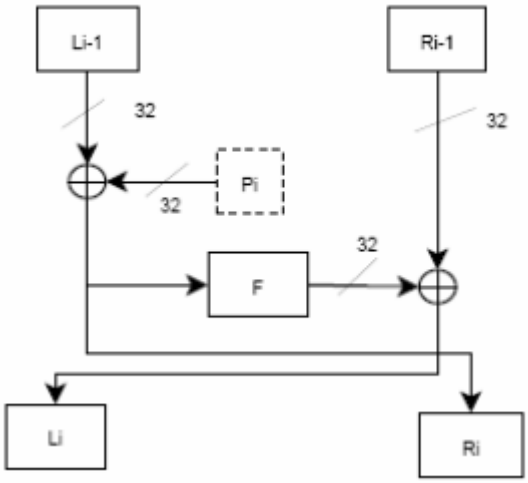
"⊕" chỉ phép XOR.

Kết quả của phép mod là để giữ 32 bit quan trọng tối thiểu. Điều này có nghĩa là phép cộng có thể hoàn thành với một dãy số 32 bit mà bỏ qua phần dư. Hình 3 thể hiện phép làm tròn với chiều dài của dãy số.





Hình 1 – Tính toán mã hóa theo thuật toán blowfish.



Hình 2 – 1 vòng tính toán của giải thuật blowfish.

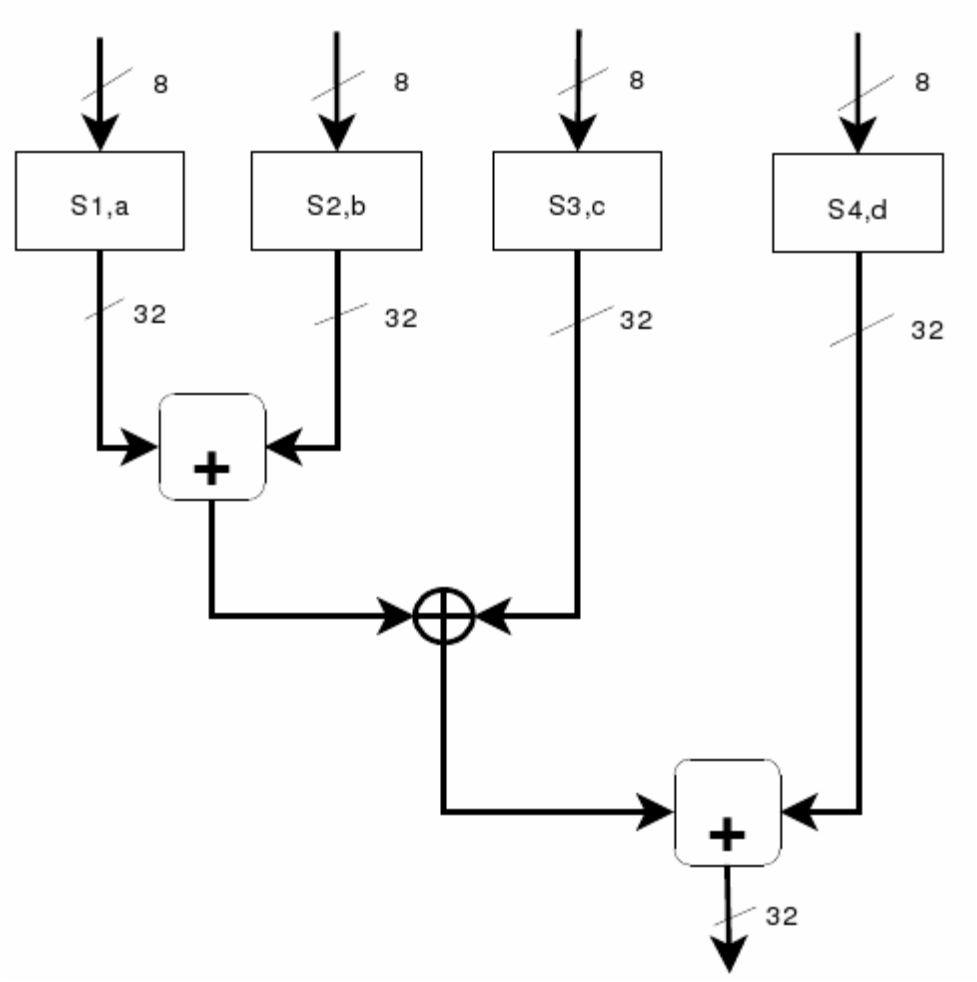
5. Tính toán khóa con Blowfish và s-boxes:

Những khóa con được sinh ra bằng thuật toán mã hóa Blowfish. Những khóa con và s-boxes được khởi tạo với con số hex của pi. Những khóa con này được thực hiện bởi phép XOR với tất cả bytes của khóa. Khóa được lặp lại tới khi tất cả 18 khóa con được thực hiện phép XOR với khóa.

Các bước tiếp theo:

1. Mã hóa tất cả những chuỗi zero với khóa được khởi tạo ở trên.
2. Thay thế P1 và P2 với kết quả của bước 1.
3. Mã hóa kết quả ở bước 1 với những khóa đã được sửa.
4. Thay thế P3 và P4 với kết quả ở bước 3.

Quá trình này được tiếp tục đến khi tất cả các khóa con và s-boxes được thay thế bởi kết quả mã hóa. Tổng số 521 vòng lặp của thuật toán mã hóa phải được hoàn thành để tính toán ra khóa con. (Tổng cộng là 4*256 + 18 giá trị sẽ được thay thế, mỗi vòng lặp tính toán 2 giá trị)



Hình 3 – Hàm F của thuật toán blowfish.

6. Phép thử vectơ Blowfish:

6.1. Sự phát sinh khóa và mã hóa:

Giả sử ta nhập khóa có giá trị 0123456789ABCDEF16. Khóa con được khởi tạo theo con số đầu tiên của pi. Những khóa con (P) được lặp đi lặp lại bởi phép XOR với khóa. Kết quả được thể hiện ở bảng 1.

Bước tiếp theo, những khóa con sẽ được tính toán bằng cách thực hiện mã hóa lặp đi lặp lại (xem chi tiết ở mục 1.5). Bảng 2 trình bày kết quả của những khóa con. Nếu chuỗi zero (0x0) được mã hóa, kết quả là 0x24594688 5754369A.

0	0x251C2FEF
1	0x0C08C53C
2	0x123ACF49
3	0x8ADBBEAB
4	0xA52A7D45
5	0xA034FC3F
6	0x090DBFFF
7	0x65E5A166
8	0x440B6481
9	0xB17BDE98
10	0xBF7723A8
11	0xBD42C183
12	0xC18F6CD0
13	0x40D79D32
14	0x3EA790D2
15	0x3CECC4F8
16	0x933590BE
17	0x00D236F4

Bảng 1 – Giá trị khởi tạo được XOR với khóa

0	0x02558D03
1	0x8BD2E05B
2	0xE60C00EA
3	0x95C270EF
4	0xF03C7555
5	0x58BA57C4
6	0xA6211A41
7	0xE731516E
8	0xF12165C7
9	0xED622043
10	0x3FF1BC45
11	0x92D1E14D
12	0x12ECBC27
13	0x638CCE1E
14	0x8D184978
15	0x330B9226
16	0x260B1EBB
17	0xC1F9262B

Bảng 2 – Kết quả cuối cùng của khóa con.