

Thao tác trong file excel

- Dùng công cụ chuyển đổi từ dạng “string” qua “Hexadecimal”

Đổi chuỗi sang mã hex			
Chuỗi	Hello wo	Hex	48656C6C6F20776F

- Có được dữ liệu từ “input” và “Key” => có được mã ( đã được mã hoá )

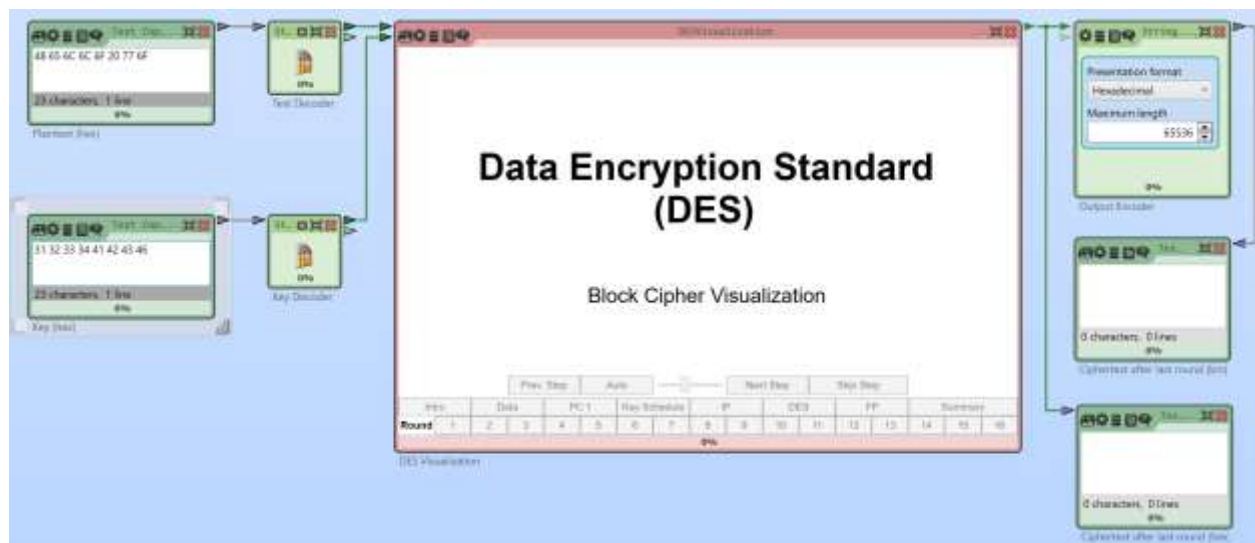
INPUT ( kí tự )	Hello wo
KEY(Mã HEX)	31323333441424346
MÃ	0A 30 66 5C 8A 5D 01 1F

Thao tác trong Cryptool

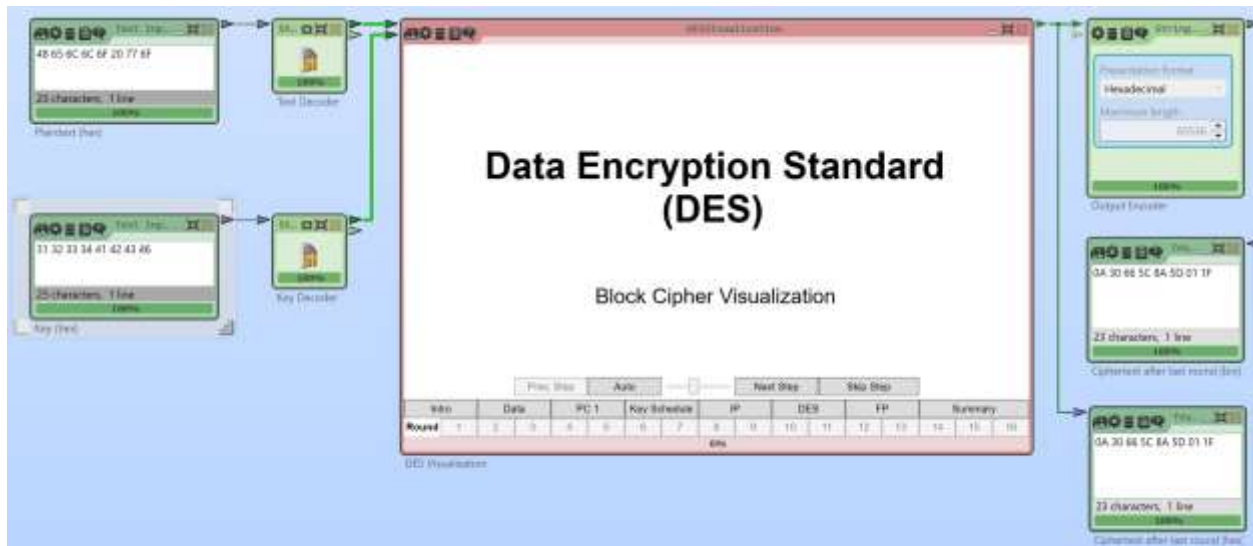
- Giao diện:



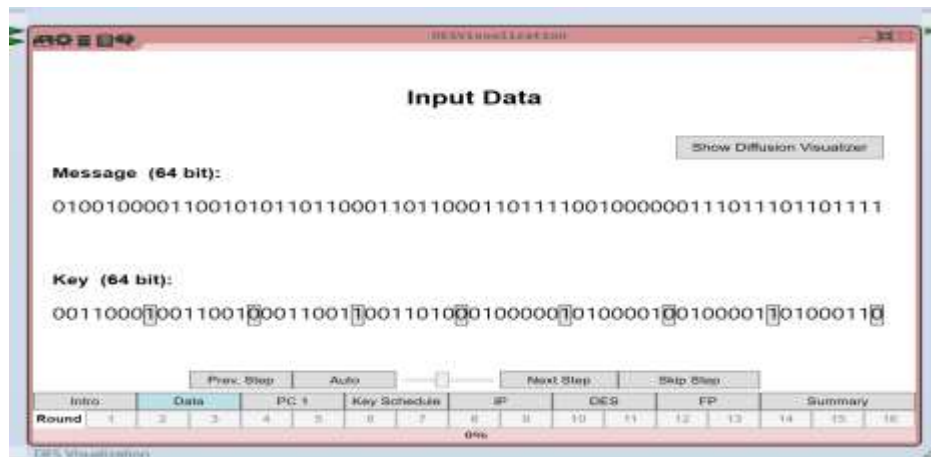
B1: lấy input đầu vào ( ở dạng Hex) đưa vào plaintext và lấy khoá chính (Key)



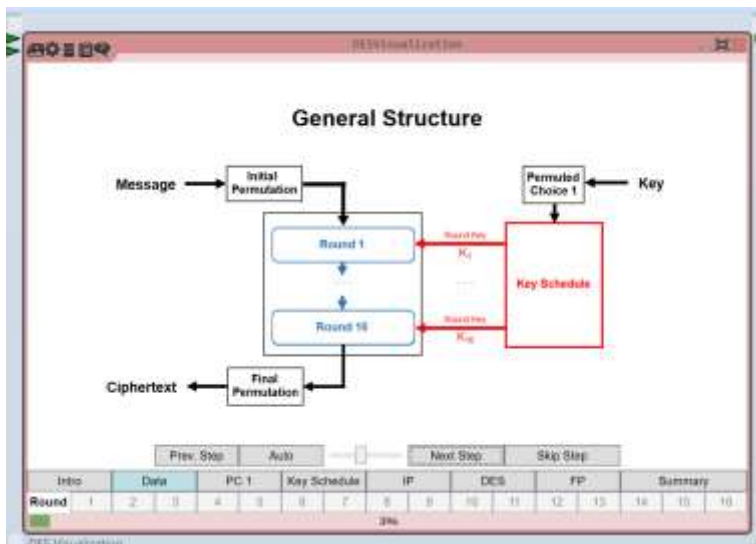
B2: Nhấn play chạy chương trình, nó sẽ trả ra mã đã được mã hoá



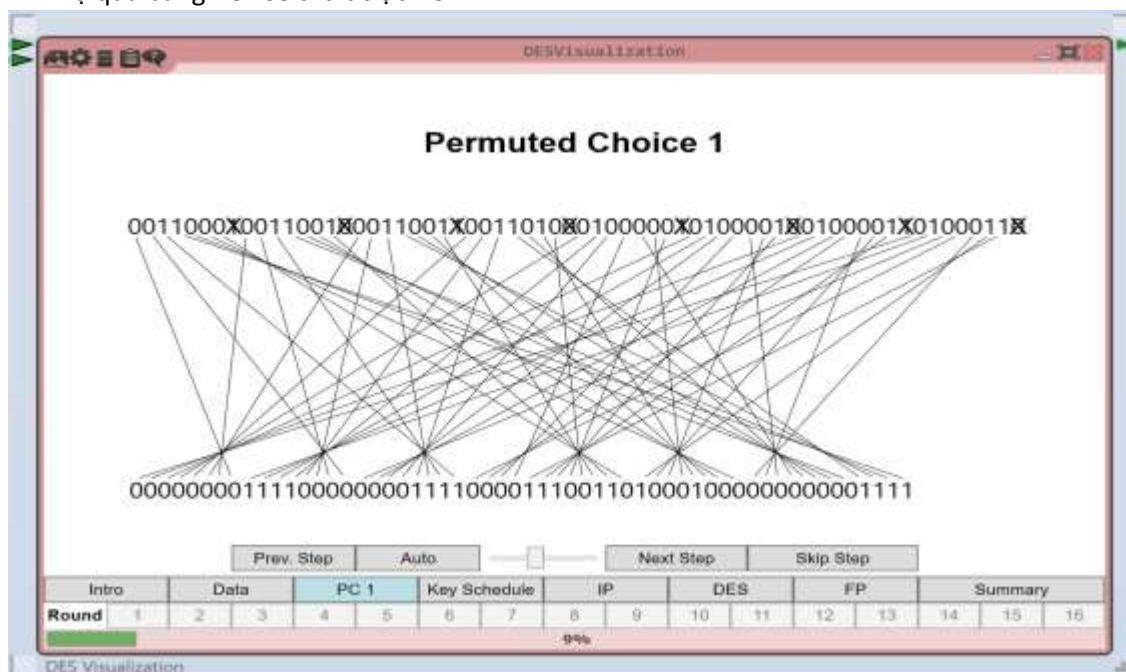
- Quan sát rõ hơn các bước
  - Đối với Key
- + dữ liệu đầu vào khi được chuyển từ Hexa sang binary



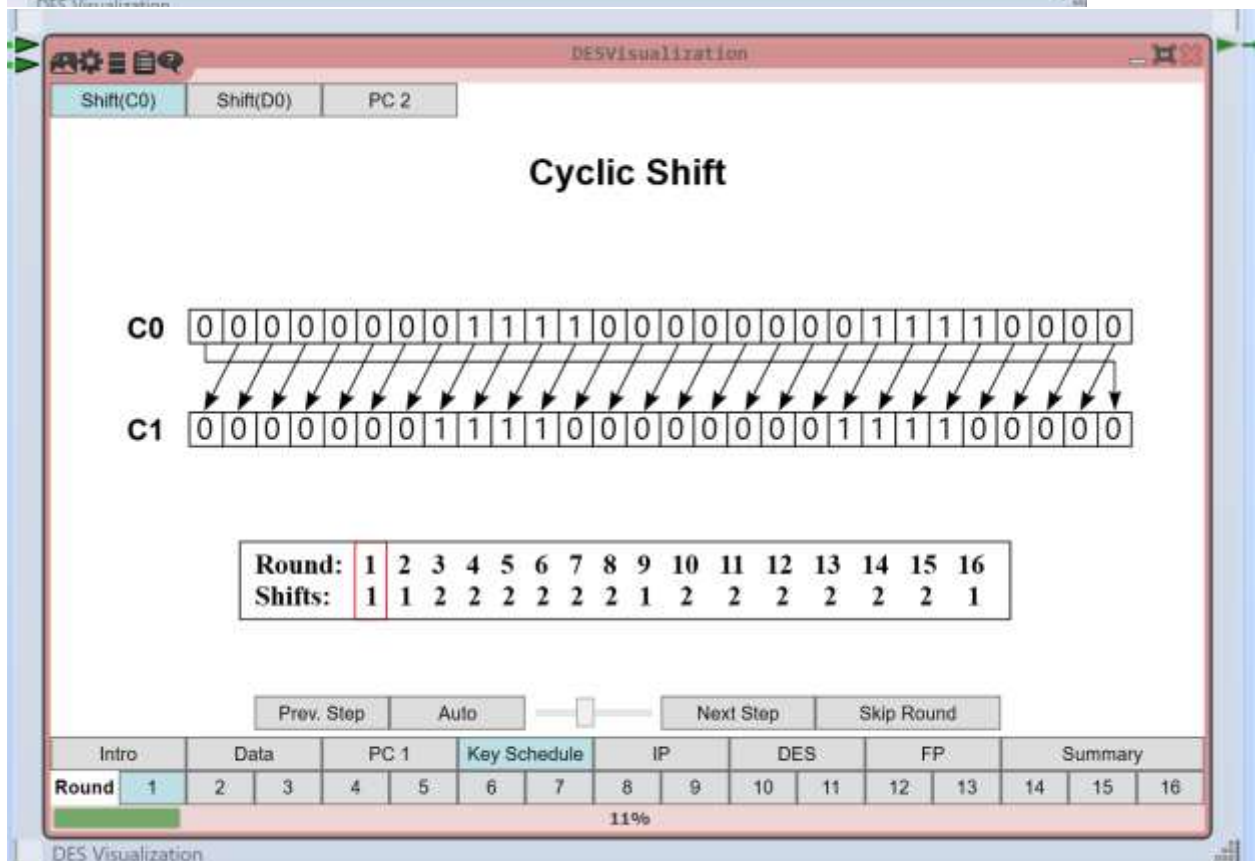
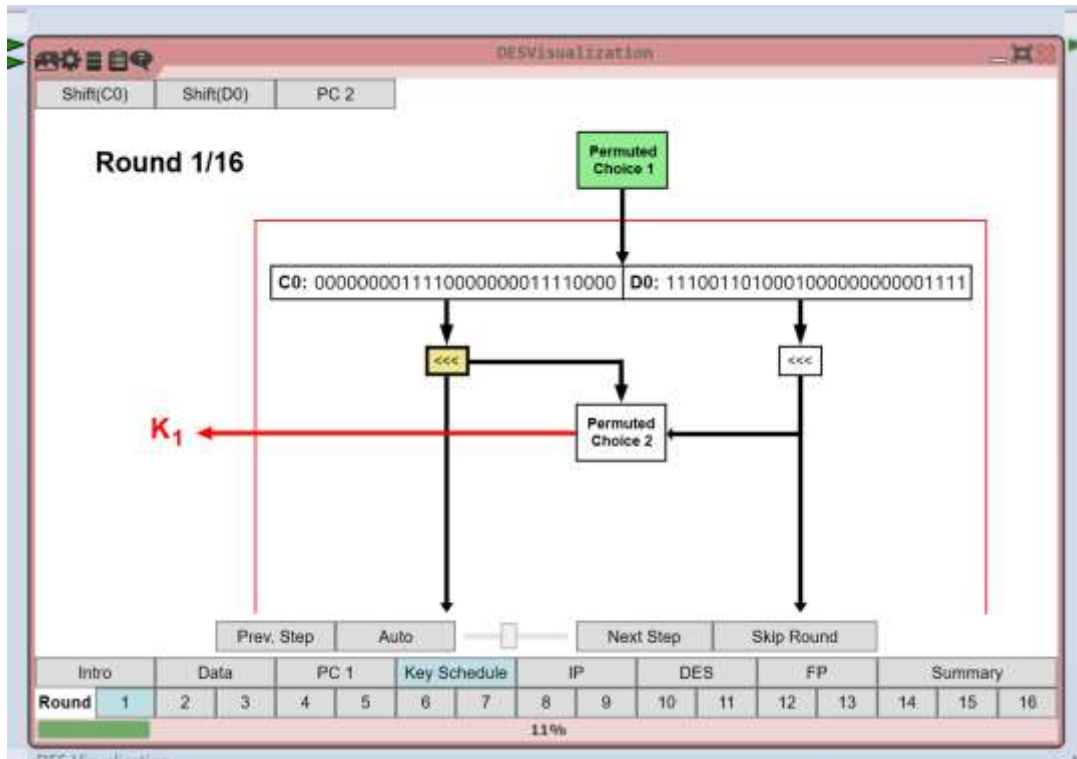
- Sơ đồ



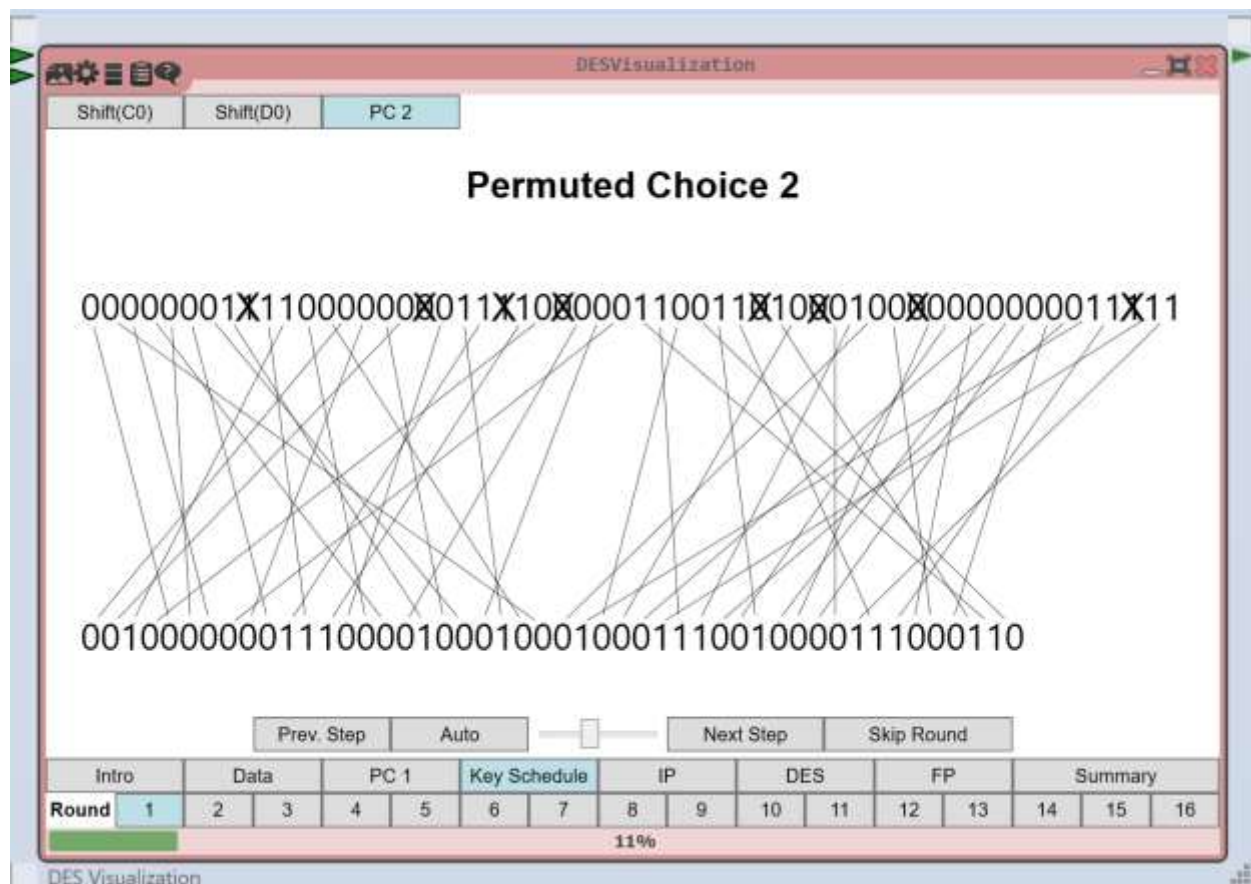
- Ảnh xạ qua bảng PC1 sẽ thu được PC1



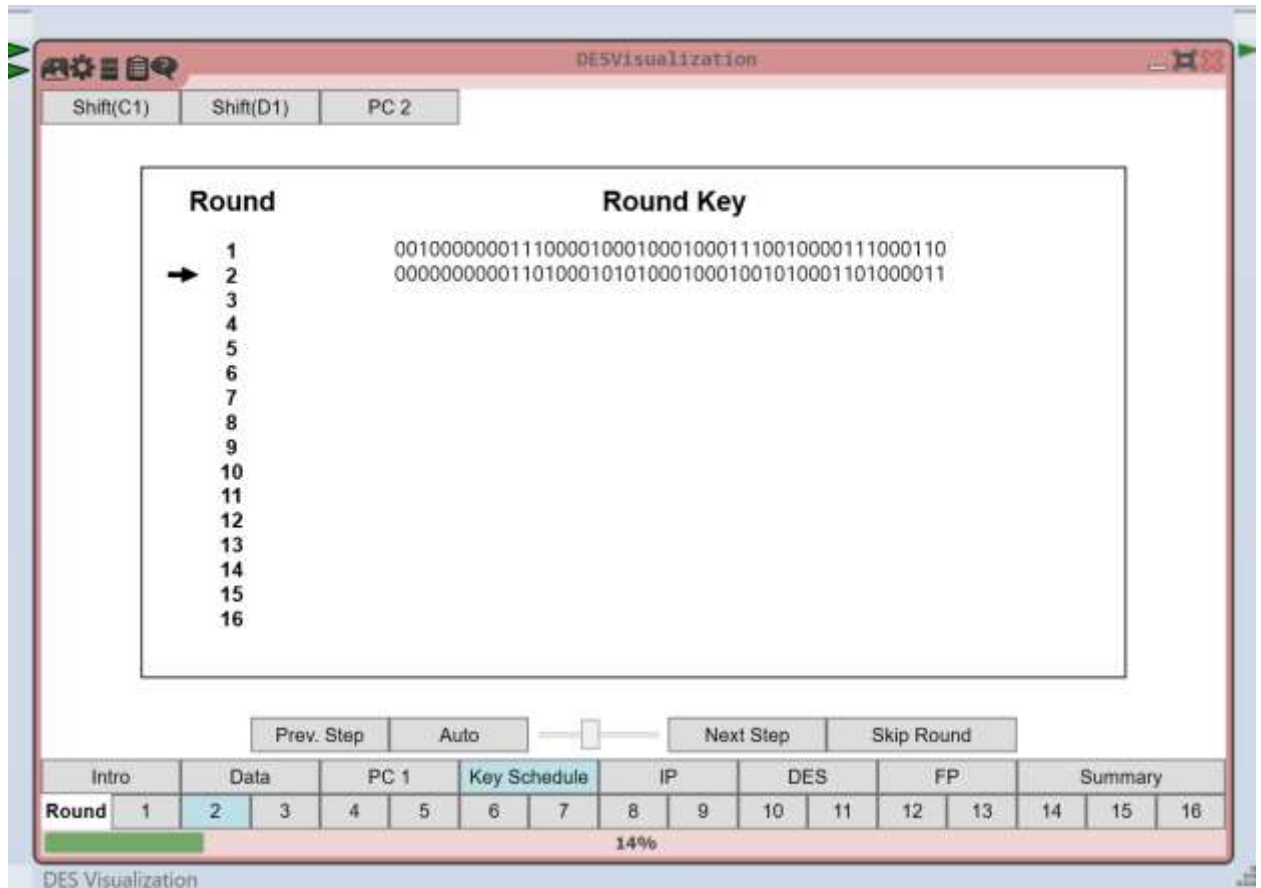
- Có PC1, dữ liệu được chia thành 2 nửa và bắt đầu dịch 1 bit cho vòng khoá 1 và vòng khoá 2 ( ở cả 2 nửa D và C)



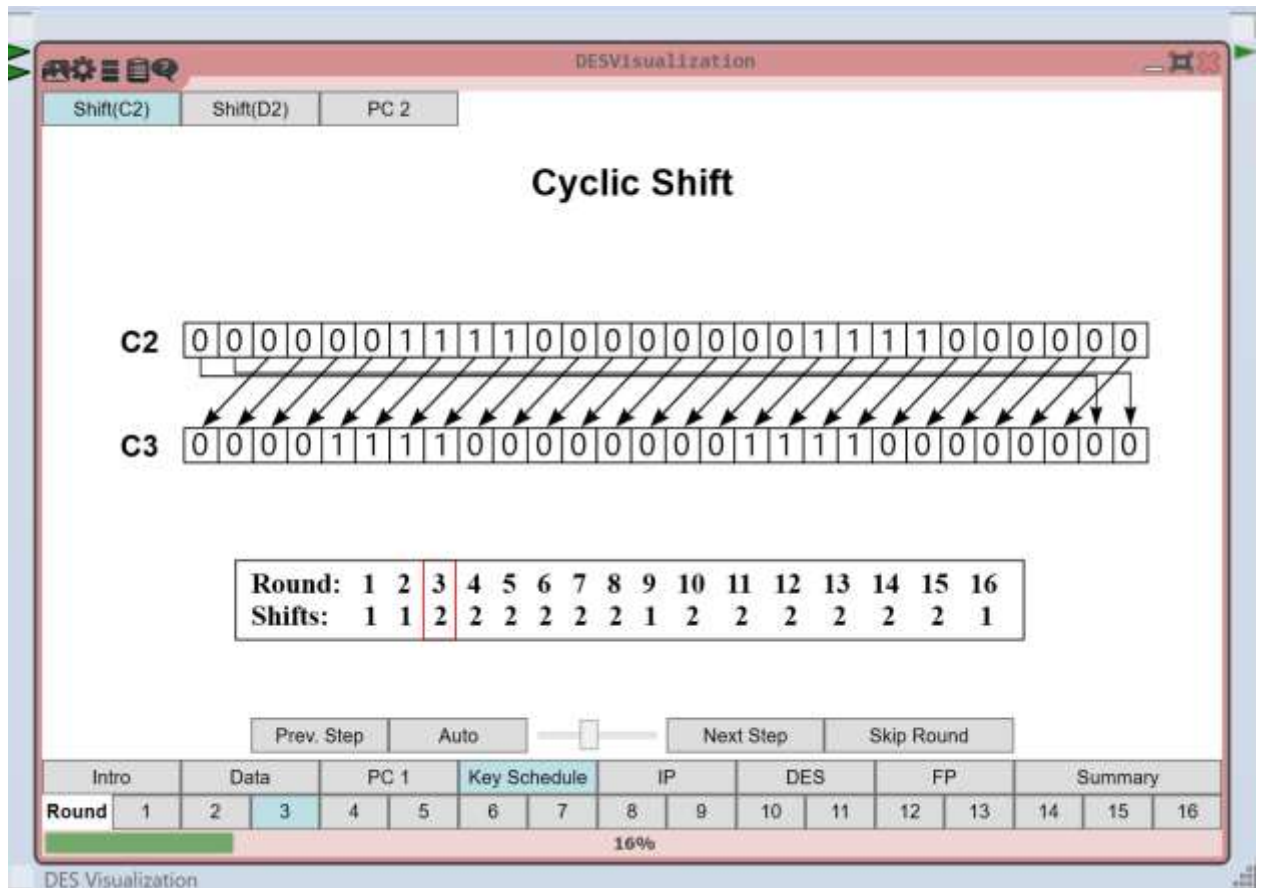
- Sau khi dịch 1 bit thì dữ liệu đi qua hoán vị PC2 sẽ thu được PC2



Sau đó tạo ra khoá con K1 và tiếp bước thứ 2 sẽ tạo ra khoá con k2

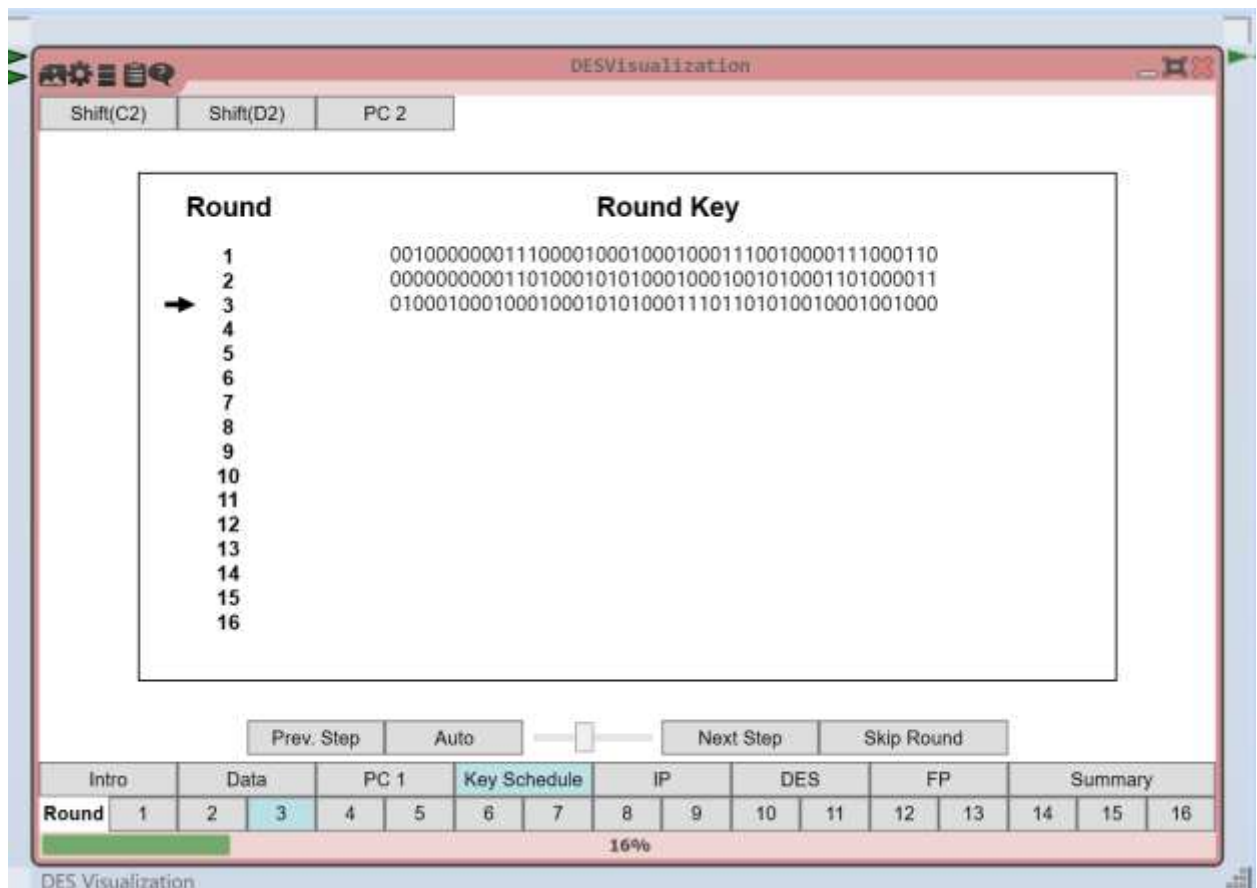


- Ở vòng thứ 3 tương tự với các bước trên nhưng khi dịch bit thì chương trình sẽ dịch 2 bit ( cho 2 nửa D và C )



- Sau đó 2 nửa được gộp lại để trải qua hoán vị PC2 và tạo ra khoá con K3





- Tiếp tục các bước trên cho tới khi có đủ 16 khoá con



DESvisualization

Shift(C15) Shift(D15) PC 2

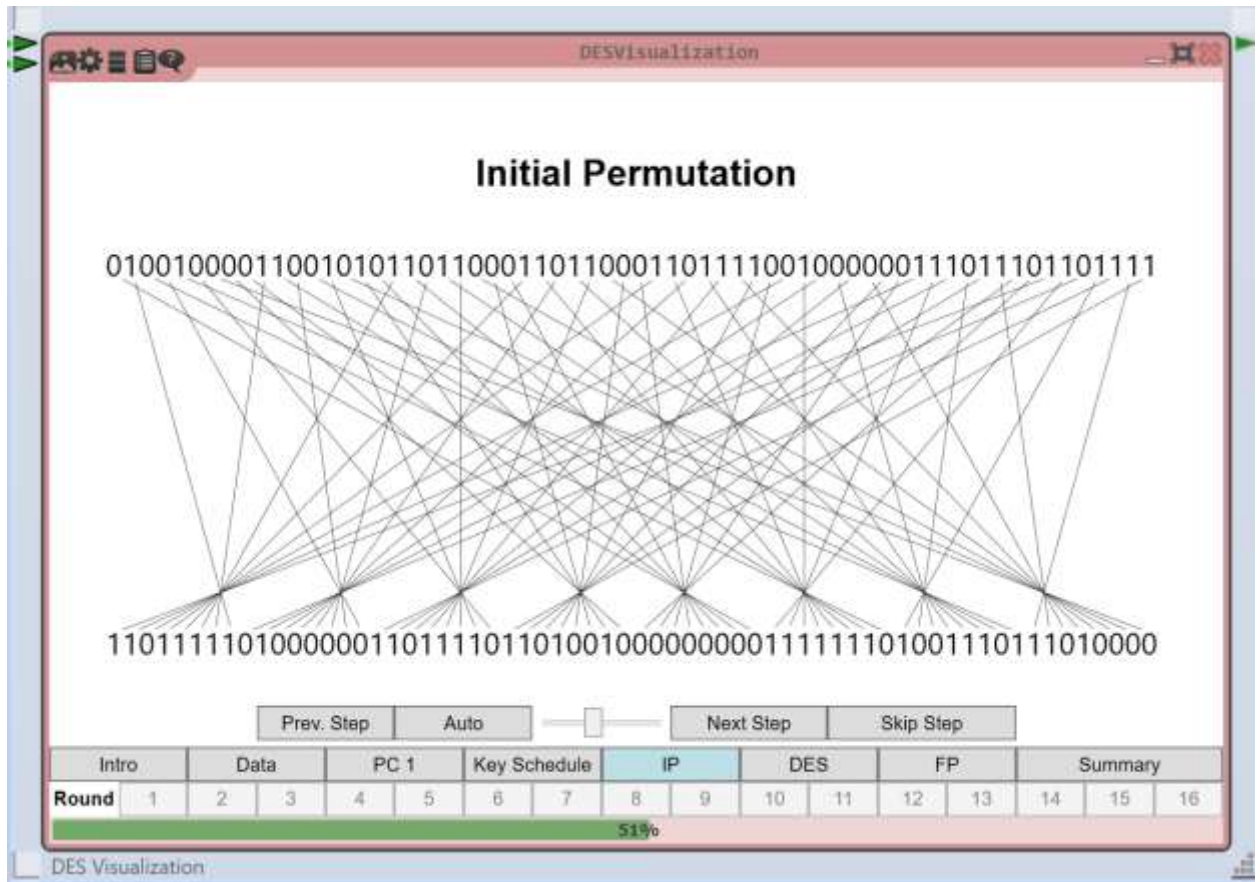
Round	Round Key
1	001000000011100001000100010001110010000111000110
2	000000000011010001010100010001001010001101000011
3	010001000100010001010100011101101010010001001000
4	0100011011000001001000000110100010010101001010
5	100010101000000100100011000011001111010000101010
6	101010010000001000001011011011000101110001100000
7	001000010001001010001000100010001100100001111010
8	000100000001100011010000100001011101111000010000
9	000100000100100001010000100010010001101000011000
10	000001000110100100010100110100010101001000110100
11	000001100010010100000101000100010000101010101100
12	010010110000010000100001100100000011100010010101
13	110010011000000010101000001000110010001010110101
14	100100001000001010001010001100110010100110000011
15	001100000001101000000010001001100000000100010111
16	001100000011101000000000101101100000000111000010

Prev. Step Auto Next Step Skip Round

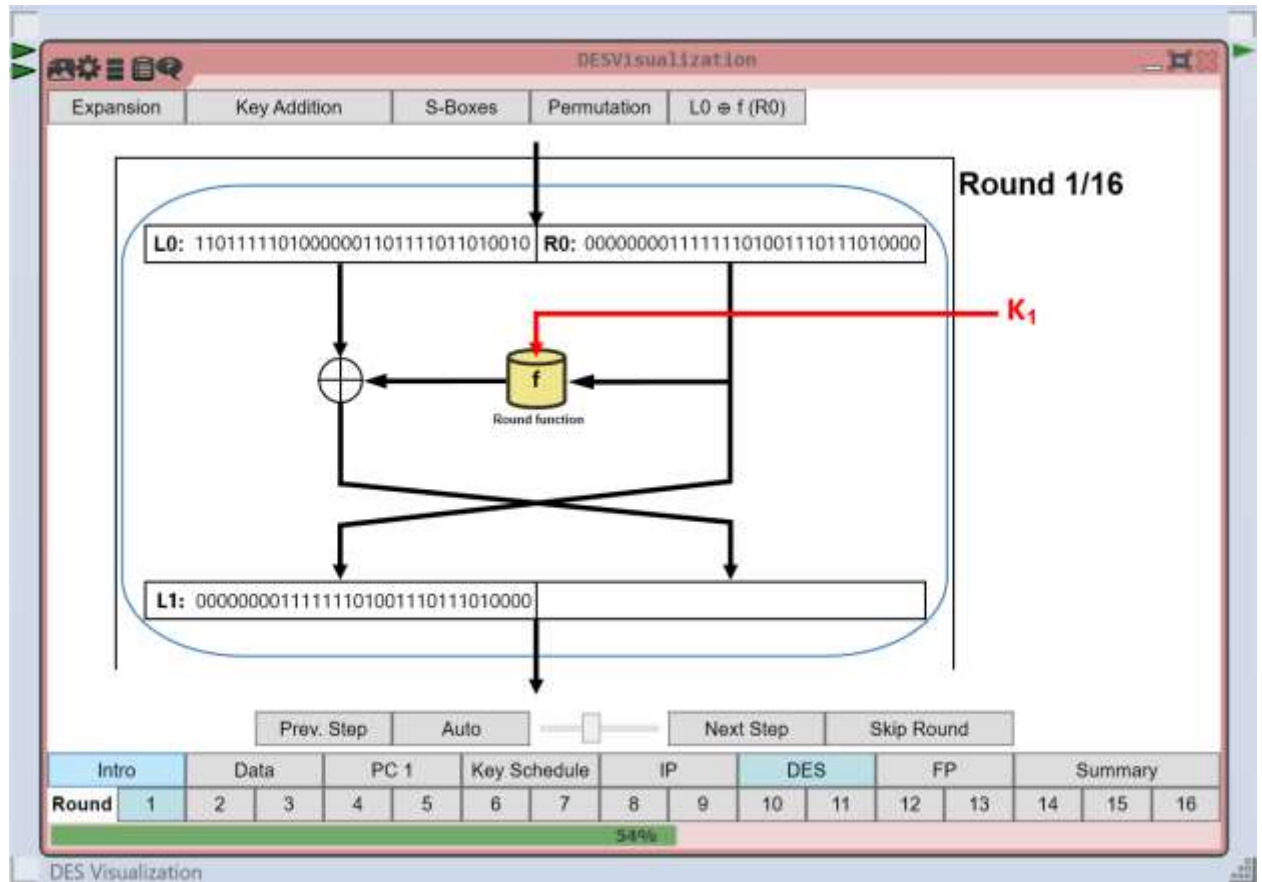
	Intro	Data			PC 1		Key Schedule		IP		DES		FP		Summary	
Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

49%

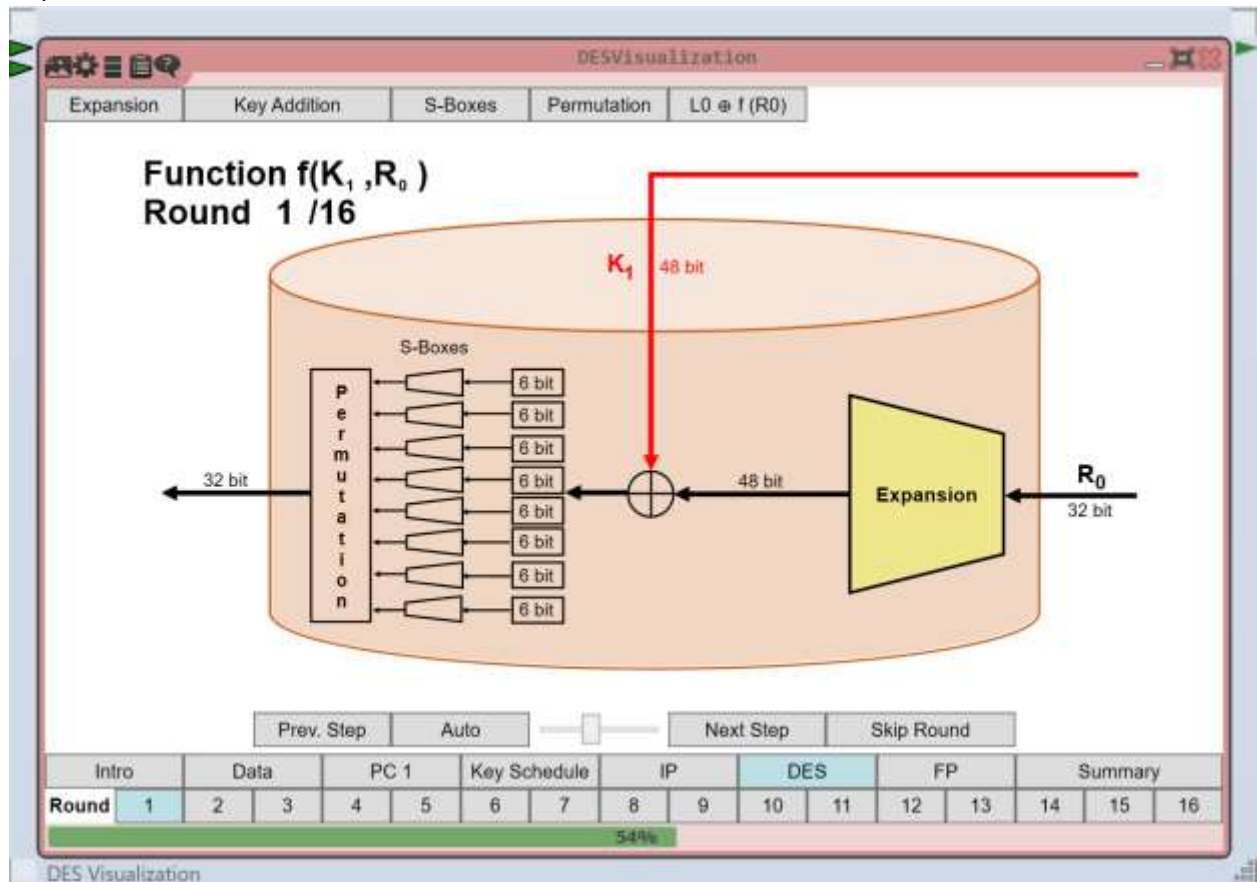
- Đối với khối đầu vào (Plain text)
- + Khối binary ban đầu sẽ đi qua hoán vị IP ,thu được IP



- Sau đó dữ liệu được chia thành 2 nửa ( trái và phải )
  1. Nửa bên phải sẽ được chuyển thẳng xuống nửa trái làm giá trị mới và nửa bên phải cũng đi vào hàm Feistel ( cục chữ f)



2. Nửa phải được mở rộng thành 48 bit , sau đó đi qua phép XOR kèm kết hợp với khoá K1 (48 bit)



- ở đây sẽ tạo ra giá trị sau phép XOR và kết quả này sẽ đi vào “ Thế S-Box “

DESVisualization

Expansion

Key Addition

S-Boxes

Permutation

L0 @ f (R0)

## Bitwise XOR Operation

**K1:** 001000000011100001000100010001110010000111000110

⊕

**Exp:** 000000000001011111111101010011111011111010100000

---

001000000010111110111001000010001001111101100110

→ **Result for S-box application**

Prev. Step

Auto

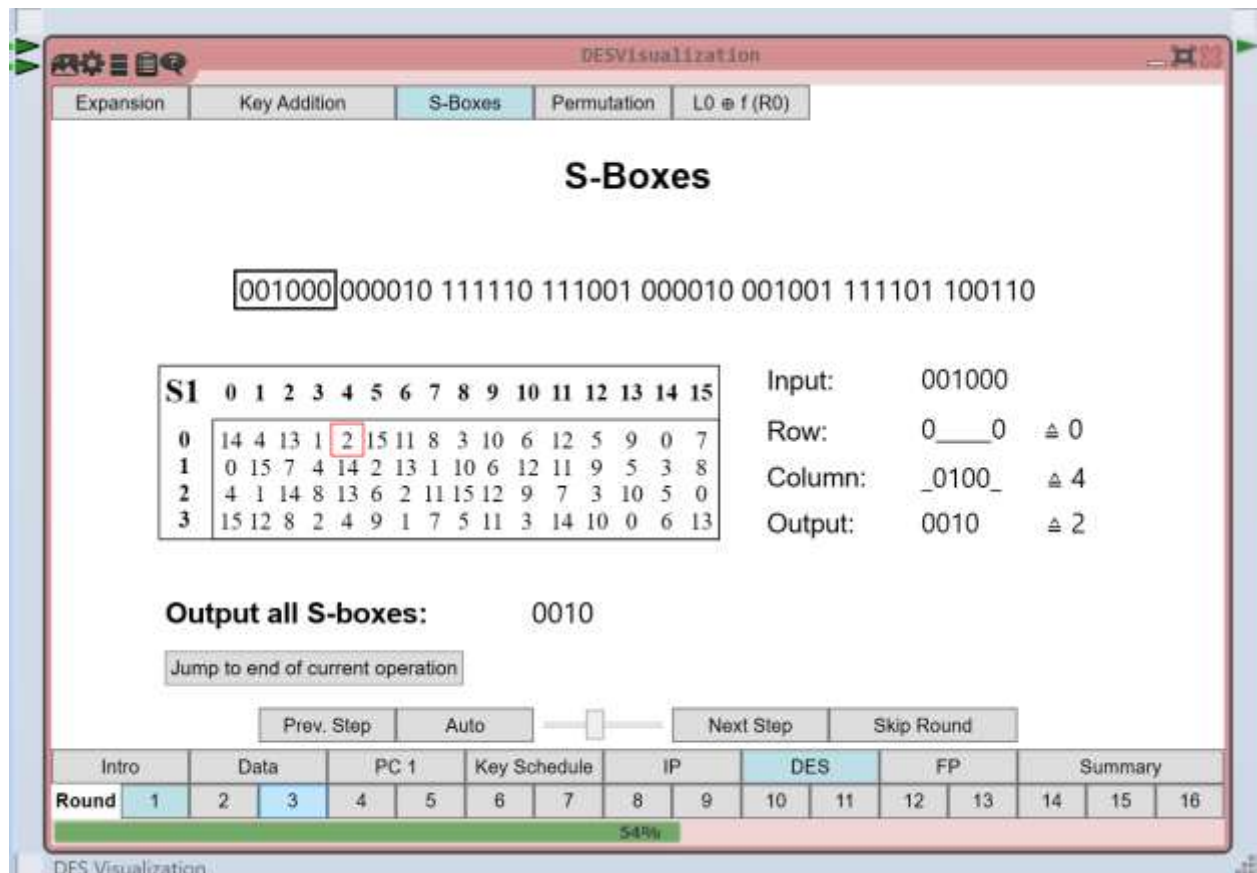
Next Step

Skip Round

	Intro	Data			PC 1		Key Schedule		IP		DES		FP		Summary	
Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

54%

3. Khối dữ liệu sau phép XOR sẽ trải qua thay thế trên S-Box

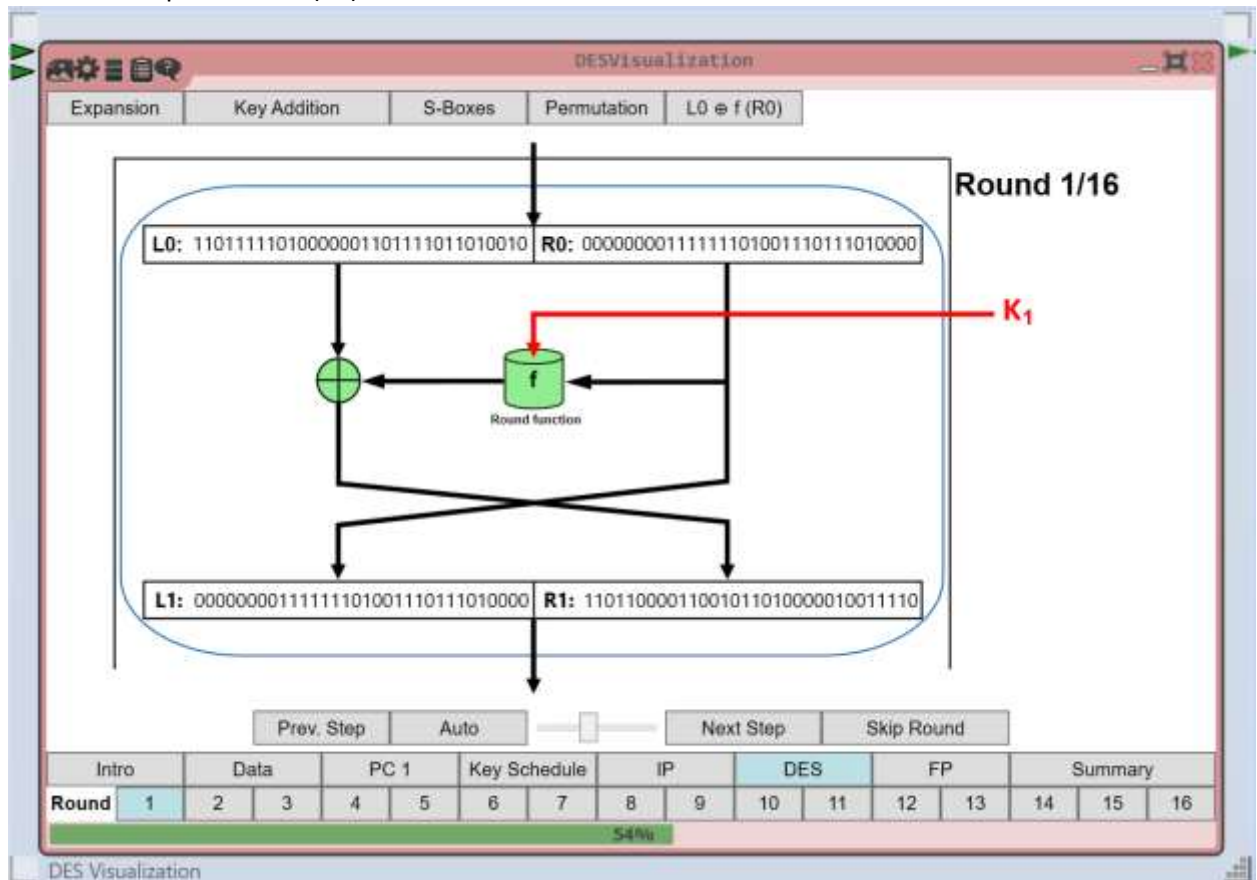


- Giải thích :
  - + Dữ liệu được chia thành các cụm 6 bit
  - + Bit đầu và Bit cuối của cụm sẽ cho biết vị trí của hàng
  - + 4 Bit giữa sẽ cho biết vị trí của cột
  - + Có được Output cho vòng 1 của S-Box là 2
- Tương tự như vậy, nó sẽ đi qua 8 vòng thay thế trong S-Box
- Sau khi thu được kết quả của 8 vòng S-Box, nó sẽ đi qua hoán vị P và cho ra 32 bit đầu ra

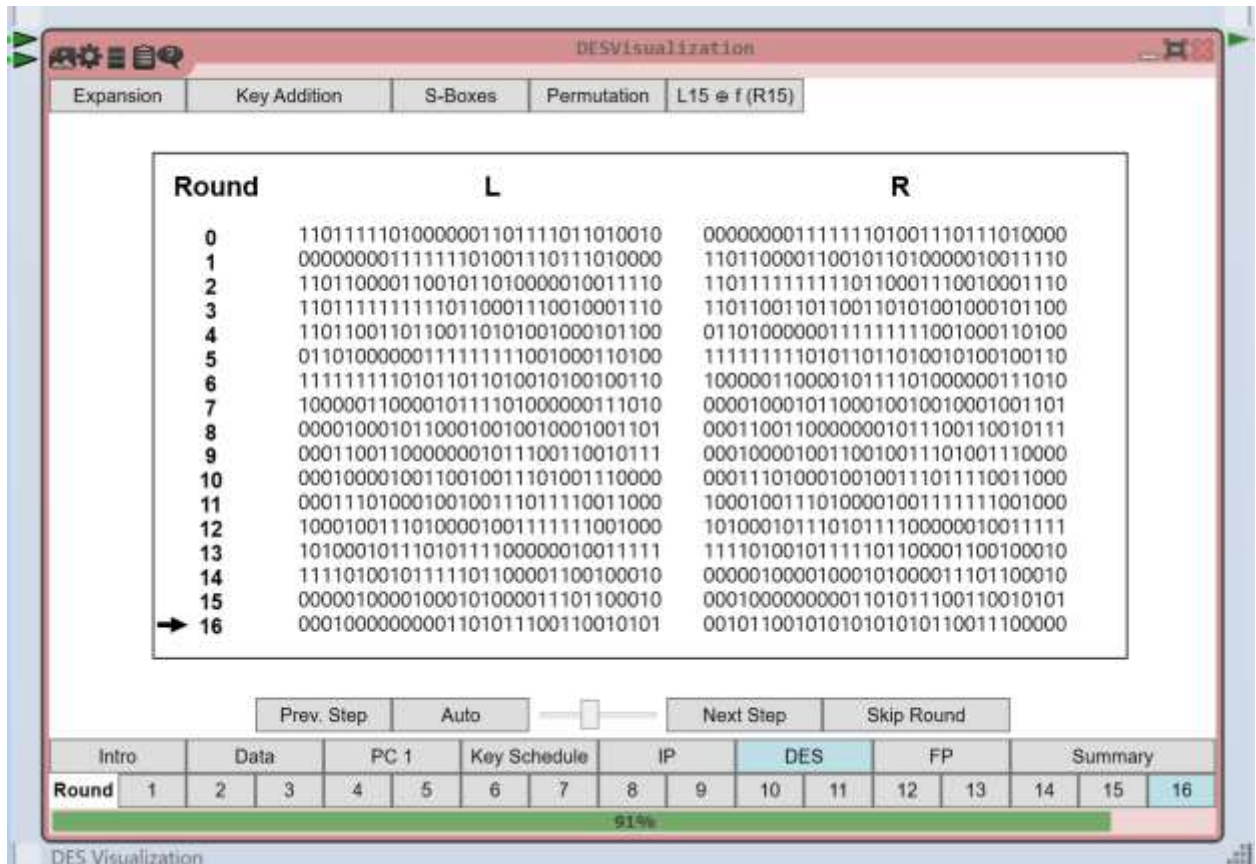




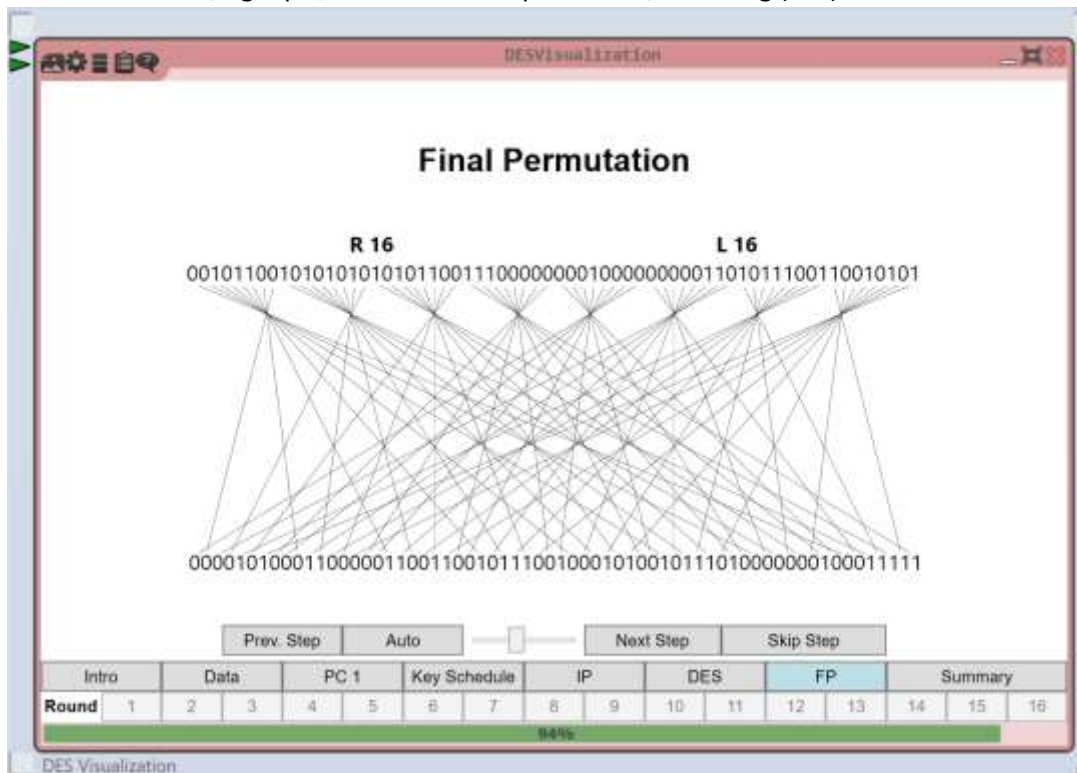
- Kết quả sau khi trải qua hàm Feistel sẽ đi tiếp tục được XOR với nửa trái ban đầu để tạo ra giá trị mới cho nửa phải ở dưới (R1)



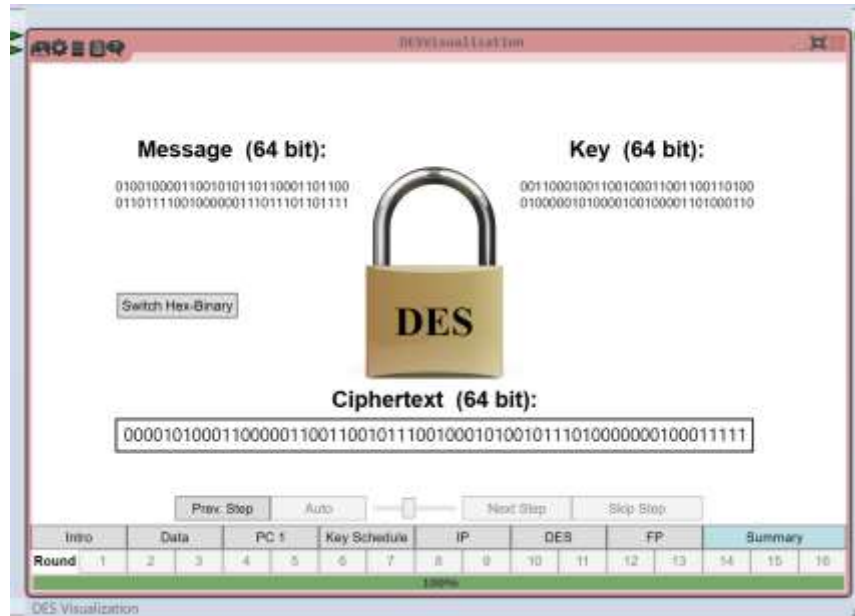
- Và tiếp tục như vậy cho đến hết 16 vòng



- Sau đó 2 nửa được ghép lại với nhau và đi qua hoán vị cuối cùng ( FP )



-Ở dạng binary



- Ở dạng Hexadecimal

