

1. permutation choice 1 (pc-1) and final permutation (inverse of pc-1) of first 16 bits
PC-1: permutes a 64-bit key to remove parity bits, leaving 56 bits
PC-2: (final permutation choice): Reduces 56-bit combined key (48 bits) to 48-bit moving keys

2. compression box: $64 \rightarrow 48$ bits
we'll simulate a design that selects 48-bits from a 64-bit input

3. Expansion box: $48 \rightarrow 64$ bits
we simulate a method to expand 48 bits to 64 by repeating some bits

```
from typing import List, Tuple, Dict, Set, Int, Str, Table, List[Str]
def permute(input_bits: Str, table: List[Str]) -> Str:
    """ permute "input_bits" according to the table """
    return ''.join([input_bits[i] for i in table])
```

#1. Permutation choices? (PPT) from 64-bit key to 56-bit key, given 16 notations

PC1 - TABLE = Σ 16 notations

57, 49, 41, 39, 25, 17, 9, 1, 11, 23, 31, 43, 55, 67, 79

1, 58, 50, 34, 2, 38, 26, 18, 10, 9, 19, 7, 3, 21

16, 29, 59, 51, 43, 35, 27

19, 11, 3, 60, 52, 44, 36

69, 55, 47, 39, 31, 23, 15

7, 62, 54, 46, 38, 30, 22

14, 6, 61, 53, 45, 37, 29

21, 19, 5, 28, 26, 12, 4

#2. Comprehension Box: comprehension box to

48 bits (custom example)

comprehension Box = [

1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]

1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]

1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]

1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]

1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]

1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]

1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]

1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]

1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]

#5. Expansion Box: Expanded 48 bits to 64

(custom example); \leftarrow μ in

Expansion - $B_{0,0,1} = \sqrt{4\pi} = 0.7746$ - best approximation

1,2,3,4,5,6,7,8,9,0-10/22/99 mon. 11:36 AM

17, 18, 19, 2652N522, 23, 24

31, 32, 33, 34, 35, 36, 37, 38, 39, 40

1953-1954. 27938423, 30, 31, 32, 33, 34.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

#Example 64 bits input (as binary string)

input_64bit = "01101101011101000161616167000111

Room 44111001161116161661611116037

Feb 2000 10:00 AM

#1. Apply PC-1

new_E6[bit] = permute (input - 64bit, PC1 -

TABLE). In the full table (not in table),

1. Print (f" 56-bit key (permuted key - 64bit))

#2. compress 64 \rightarrow 48bit (MBP)
compressed - 48bit = permuted input - 64
bit, COMPRESSION-BOX)

print (f" compressed 48-bit output
{compressed - 48bit} ")

#3. Expand 48 \rightarrow 64 (MBP)
expanded - 64bit = permuted (compressed

- 48bit, EXPANSION-BOX)

print (f" expanded 64bit output : {

(print (expanded - 64bit)))

10000010101010001011101011010 = f1d1032b99

1001111010010101101100110001 = 1111032b99

100000010101010001011101011010 = f1d1032b99

- 109 f1d1032b99 1111032b99 1111032b99

100000010101010001011101011010 = f1d1032b99

100000010101010001011101011010 = f1d1032b99