This is my encryption algorithm.

def rule9b0(b):

#get indexed

row = int(b[0])

col = int(b[1:], 2)

matrix = [['101','010', '001', '110', '011', '100', '111', '000'],

['001', '100', '110', '010', '000','111', '101', '011']

]

return matrix[row][col]

def rule9b1(b):

#get indexed

row = int(b[0])

col = int(b[1:], 2)

matrix = [['100','000','110','101','111','001','011','010'],

['101','011','000','111','110','010','001','100']]

return matrix[row][col]

# we need to convert the text into bits

def string2binary(text):

return ''.join(f"{ord(c):08b}" for c in text)

def binary2string(text):

return ''.join(f"{ord(c):08b}" for c in text)

def splitblock(block):

Lr = block[:6]

Rr = block[6:]

return Lr, Rr

def expand\_miniblock(b):

return b[0] + b[1] + b[3] + b[2] + b[3] + b[2] + b[4] + b[5]

def xor(a, b):

res = int(a, 2) ^ int(b, 2)

return f"{res:08b}"

def encrypt(text, key, R):

text\_encr = ''

#Rule 1.

text\_bin = string2binary(text)

if (len(text\_bin) % 12 != 0):

raise Exception(f'Rule 1 not respected.')

#Rule 2

key\_bin = string2binary(key)

if (len(text\_bin) < 8):

raise Exception('Rule 2 not respected')

#rule3

#we have some blocks ...

for bnum in range(len(text\_bin) // 12):

i = bnum

#define the block

from\_ = 0 + 12\*bnum

to\_ = 12 \* (bnum + 1)

block = text\_bin[from\_:to\_]

#rule4

for r in range(R):

#rule5

Lr, Rr = splitblock(block)

#rule6

Rr\_expanded = expand\_miniblock(Rr)

#Rule7

curr\_key = key\_bin[(i\* R + r) : ((i\* R + r)+8)]

Rr\_exp\_xor\_key = xor(Rr\_expanded, curr\_key)

#Rule8

Rr\_exp\_xor\_key\_0 = Rr\_exp\_xor\_key[:4]

Rr\_exp\_xor\_key\_1 = Rr\_exp\_xor\_key[4:]

#Rule9

Rr\_exp\_xor\_key\_0\_conv = rule9b0(Rr\_exp\_xor\_key\_0)

Rr\_exp\_xor\_key\_1\_conv = rule9b1(Rr\_exp\_xor\_key\_1)

#Rule10

Rr\_sboxes = Rr\_exp\_xor\_key\_0\_conv + Rr\_exp\_xor\_key\_1\_conv

if len(Rr\_sboxes) != 6:

raise Exception("Error on Rule 10")

#Rule11

Rr\_alt = xor(Lr, Rr\_sboxes)[2:]

#Rule12

block = Rr + Rr\_alt

#Rule13

#end of step

#append the result.

text\_encr += block

return text\_encr