

Digital Play

DESCRIPTION: My friend made this encryption circuit to encrypt the text. Could you help to decrypt that text.

AUTHOR : 0xRakesh Kumar

There are two files: encrypt.dig and enc.txt.

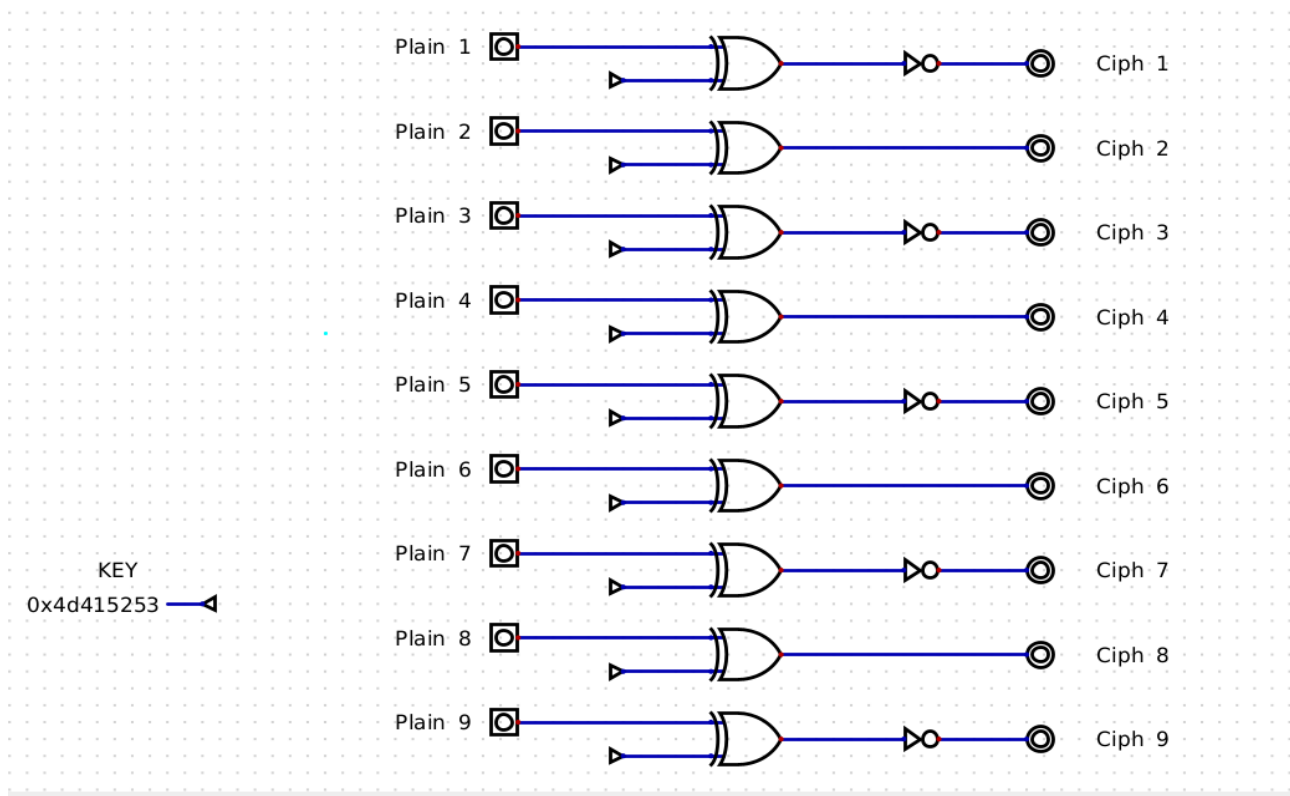
enc.txt:

```
→ Digital Play cat enc.txt
00110110111111100000011000101 100001000000100000011000010101 001001111110101001110011001011 1111100000101010000110
100010000 011011111011001110111011011001 1111100000101010110011100001100 010011111011001100100011110011 1100001101
100001011101100110 0000010111100111100100011010001
→ Digital Play _
```

It look like a binary number.

encrypt.dig :

I just search about **.dig** file extension. Then I finally found one application for open it. (<https://github.com/hneemann/Digital>).



They use a logic gates for encryption.

Process: Xor the plain text with xor key (0x4d415253) and inverse the even number plain text.

Let make an decrypt script :

1. Inverse the even number encrypt text.
2. Change the binary number to integer number.
3. Xor the integer number with xor key(0x4d415253).
4. Then unpack the hex value with the help struct module.
5. Finally change the byte text to string.

```
1  #!/usr/bin/python
2  import struct
3
4  enc_value = ['00110110111111100000011000101', '100001000000100000011000010101', '001001111110101001110011001011', '1']
5
6  inverse_value = []
7
8  flag = []
9
10 xor_key = 0x4d415253
11
12 mod = 0
13
14 for i in enc_value:
15     if mod%2 != 0:
16         inverse_value.append(i)
17     else:
18         inverse_value.append(''.join(['1' if j=='0' else '0' for j in i]))
19     mod += 1
20
21 integer_value = [int(i,2) for i in inverse_value]
22
23 xor_value = [i ^ xor_key for i in integer_value]
24
25 flag_byte = [struct.pack('I',i)[::-1] for i in xor_value]
26
27 flag = [i.decode('utf') for i in flag_byte]
28
29 print(''.join(flag))
30
31
32
```

Run the script:

[Rakesh@TamilCTF](#) → `python3 decrypt.py`

TamilCTF{D1g1T_CiRCu1T5_aRe_AwE50Me}

