

Electronic **C**oloring **B**ook



Seen in POC || GTFO 0x05

PoC || GTFO;
addressed to the
INHABITANTS
of
EARTH
on the following and other
INTERESTING SUBJECTS
written for the edification of
ALL GOOD NEIGHBORS



August 10, 2014

- | | |
|---|---|
| 5:2 A Sermon Celebrating Hacker Privilege | 5:8 A Second RDRAND Backdoor |
| 5:3 Electronic Coloring Books | 5:9 Cisco KVM Exploits |
| 5:4 Reflecting the Page Tables over PCI Express | 5:10 Shellcode that is its own NOP Sled |
| 5:5 How to make a Flash PDF Polyglot | 5:11 Rosetta Stone for SWF in ASCII |
| 5:6 SMP in 512 Bytes | 5:12 Polyglots from SHA1 Collisions |
| 5:7 PCIe over USB | 5:13 Ben Nagy's Latest Poem |

LAS VEGAS, NV:

Published at Considerable Financial Loss by the
Tract Association of PoC||GTFO and Friends,
to be Freely Distributed to all Good Readers,
and to be Freely Copied by all Good Bookleggers.



€0, \$0, £0. Самиздат. pocorgtfo05.pdf.

Soon in POC || GTFO 0x06

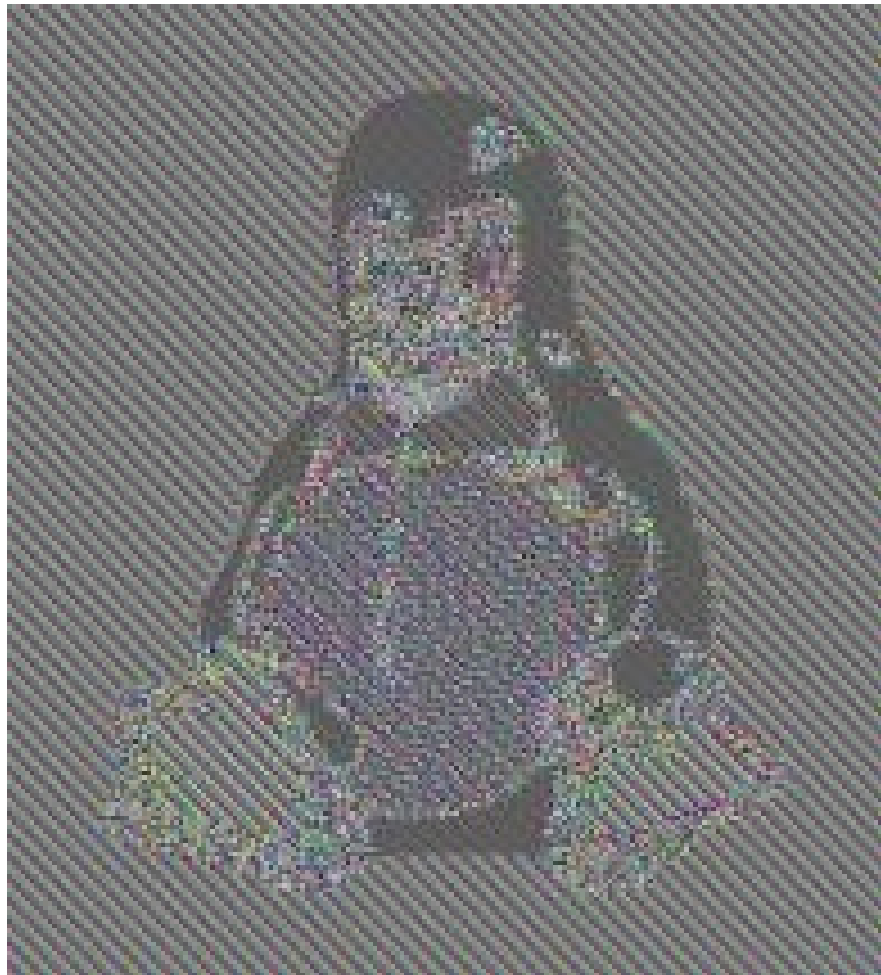


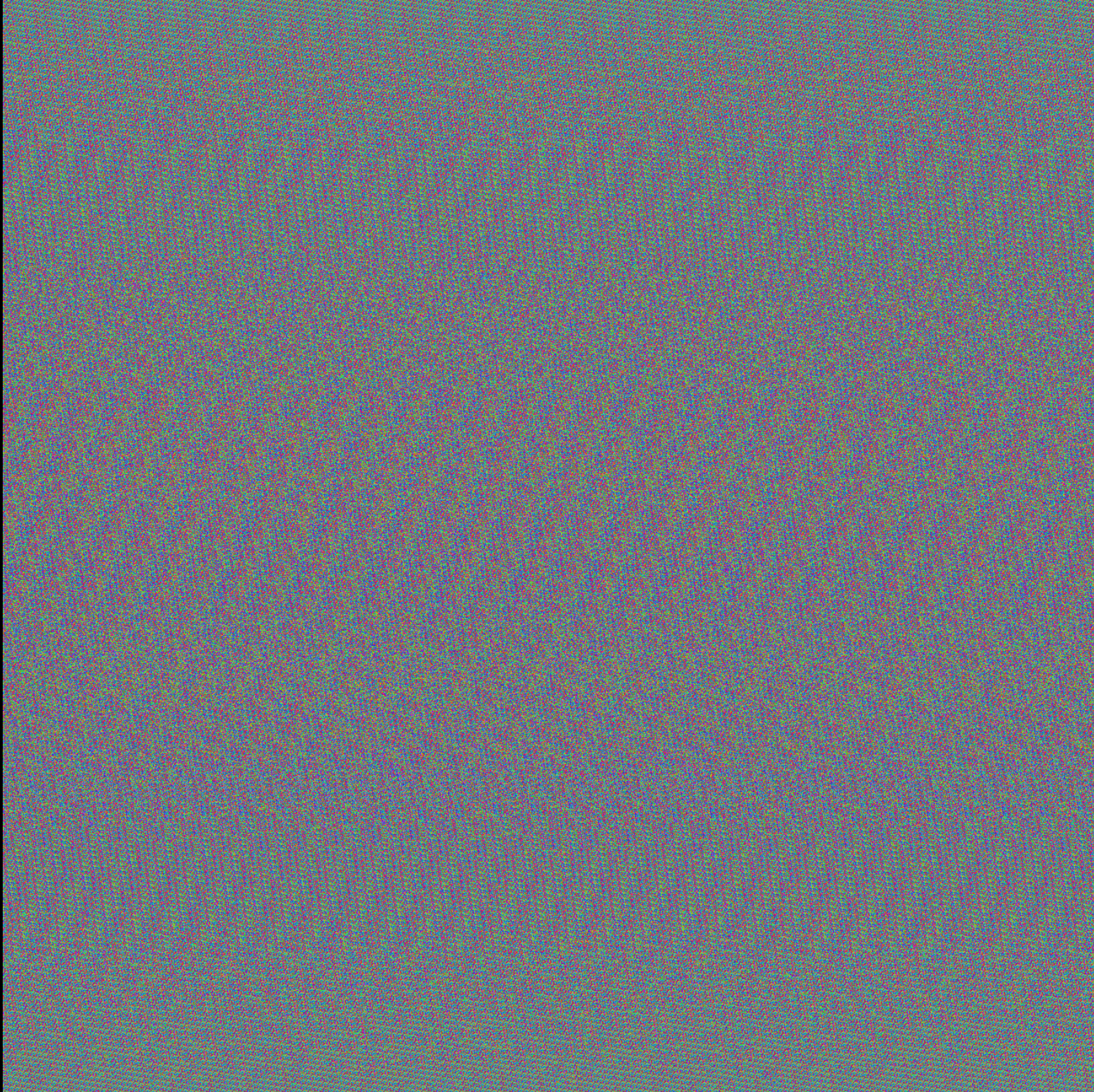
- 79 -



ECB mode is bad

Because you can see the penguin...





What can we do?

- **Stats on ECB blocks (16-byte for AES-ECB)**

c1b108f9b8cb7c020b992ea48d946a78	10018
2caef1297f191eeb7c086058de486e38	10001
5c0ce2b870019e78be581e7777988477	9906
f3f8e5ea5fbafe940ef5002f83ddd73e	9477
16eda065a407fab91b5e3ec58c390bbc	9296
3087b683a09e9663b5a5fb9b83904fcc	9224
9ce907fc9e9ae7a32064f5c49a8d3439	8238
7b1c0506a9c16aaa8176d949089c6056	8126
6a3d8e4660f8f0b7e11cce7c4f3f7fad	8081
...	
*****	24221

What can we do?

- Paint top ECB blocks with uniform colors
- Paint remaining ECB blocks in black

c1b108f9b8cb7c020b992ea48d946a78	10018	->	#FF	#FF	#FF
2caef1297f191eeb7c086058de486e38	10001	->	#28	#CC	#8A
5c0ce2b870019e78be581e7777988477	9906	->	#28	#CC	#63
f3f8e5ea5fbafe940ef5002f83ddd73e	9477	->	#28	#CC	#50
16eda065a407fab91b5e3ec58c390bbc	9296	->	#CC	#A8	#28
3087b683a09e9663b5a5fb9b83904fcc	9224	->	#CC	#75	#28
9ce907fc9e9ae7a32064f5c49a8d3439	8238	->	#42	#28	#CC
7b1c0506a9c16aaa8176d949089c6056	8126	->	#28	#CC	#3D
6a3d8e4660f8f0b7e11cce7c4f3f7fad	8081	->	#CC	#28	#3C
...					
*****	24221	->	#00	#00	#00

What can we do?

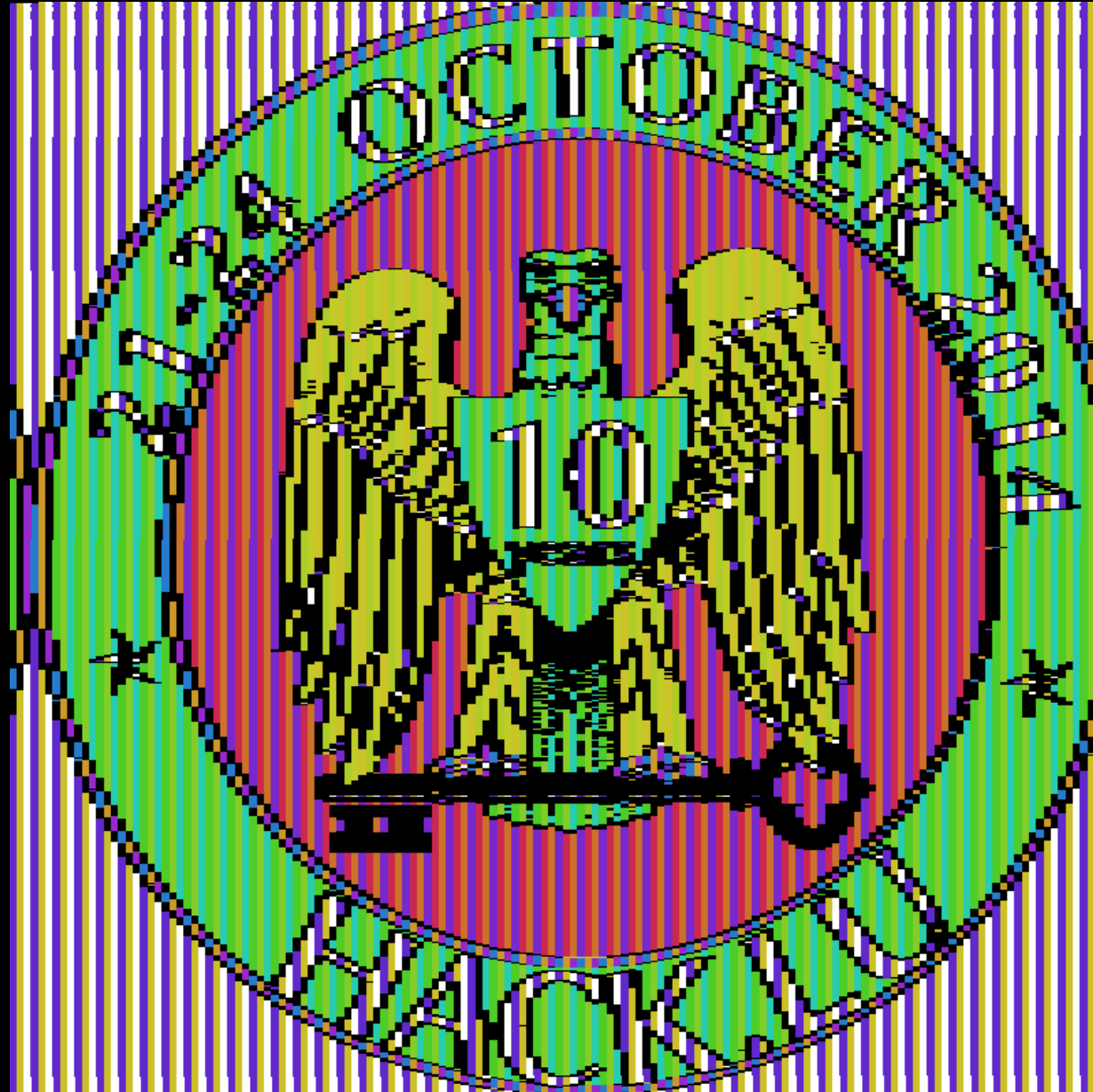
- **Guess automatically correct ratio
by correlation between adjacent lines**

See <https://github.com/doegox/ElectronicColoringBook>

```
$ ElectronicColoringBook.py test.bin
```




```
$ ElectronicColoringBook.py test.bin  
-p 3
```



Stripes?

AABBCCAABBCCAABBCCAABBCCAABBCCA	81E49040C91E64A8F2EB52EB313EADF4
BBCCAABBCCAABBCCAABBCCAABBCCAABB	769B3981E49040C9164A83B6CBFB12BF
CCAABBCCAABBCCAABBCCAABBCCAABBCC	12B4502017A19C0EB313EADF47638FB2
AABBCCAABBCCAABBCCAABBCCAABBCCA	81E49040C91E64A8F2EB52EB313EADF4
BBCCAABBCCAABBCCAABBCCAABBCCAABB	769B3981E49040C9164A83B6CBFB12BF

etc.

```
$ ElectronicColoringBook.py test.bin  
-p 3 -g 3 -o 3
```




```
$ ElectronicColoringBook.py test.bin  
-p 3 -g 3 -o 3 -P
```

```
'#ffffff#ffffff#ffffff#ffffff#ffffff#ffffff  
#000000'
```



```
$ ElectronicColoringBook.py test.bin  
-p 3 -g 3 -o 3 -P  
'#000000#ffffff#ffffff#ffffff#ffffff#ffffff  
#000000'
```



```
$ ElectronicColoringBook.py test.bin  
  -p 3 -g 3 -o 3 -P  
' #000000#ffffff#134471#ffffff#ffffff#ffffff  
  #000000'
```




```
$ ElectronicColoringBook.py test.bin  
-p 3 -g 3 -o 3 -P  
'#000000#ffffff#134471#886035#e0ae37#a39f97  
#000000'
```



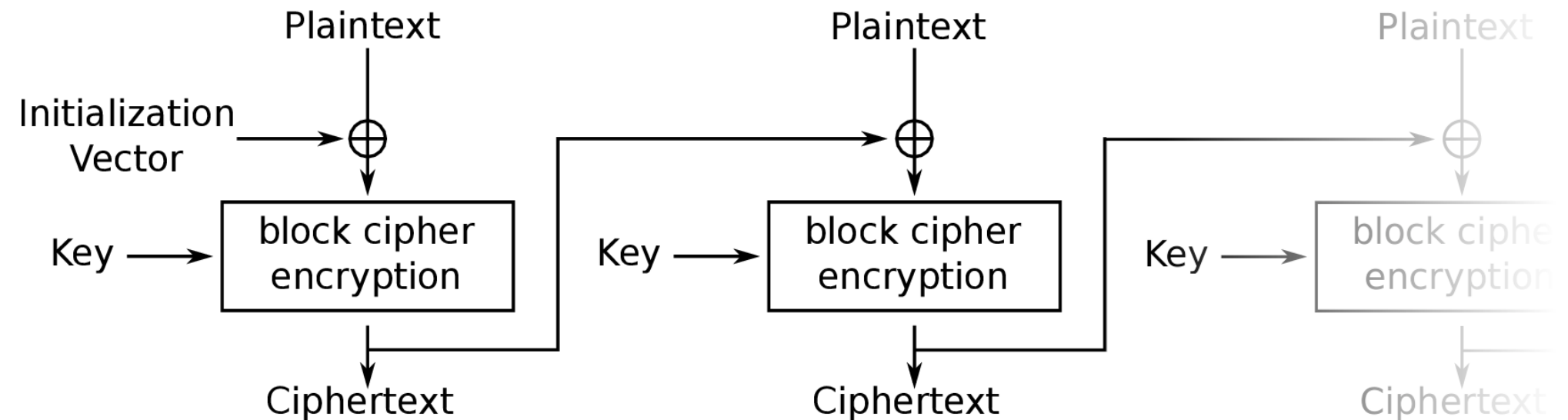
```
$ ElectronicColoringBook.py test.bin  
  -p 3 -g 3 -o 3 -P  
' #000000#ffffff#134471#886035#e0ae37#a39f97  
  #000000 '
```



AES 128

What about CBC mode?

Sneak preview of
POC || GTFO 0x06
(don't tell Travis)



Angecryption by Corkami

$$\text{AES}(\text{Image of Theresa May holding a smartphone}) = \text{Image of a man in a military uniform}$$

DEC (Google) = 

$$\text{DEC}(\text{Google}) = \text{[redacted]}$$

$$\text{[redacted]} + \text{DuckDuckGo} = \text{[redacted] DuckDuckGo}$$

$$\text{DEC}(\text{Google}) = \text{[noise]}$$

$$\text{[noise]} + \text{Duck} = \text{Duck}$$

$$\text{ENC}(\text{[noise] Duck}) = \text{Google}$$

$$\text{DEC}(\text{Google}) = \text{[noise]}$$

$$\text{[noise]} + \text{Duck} = \text{Duck}$$

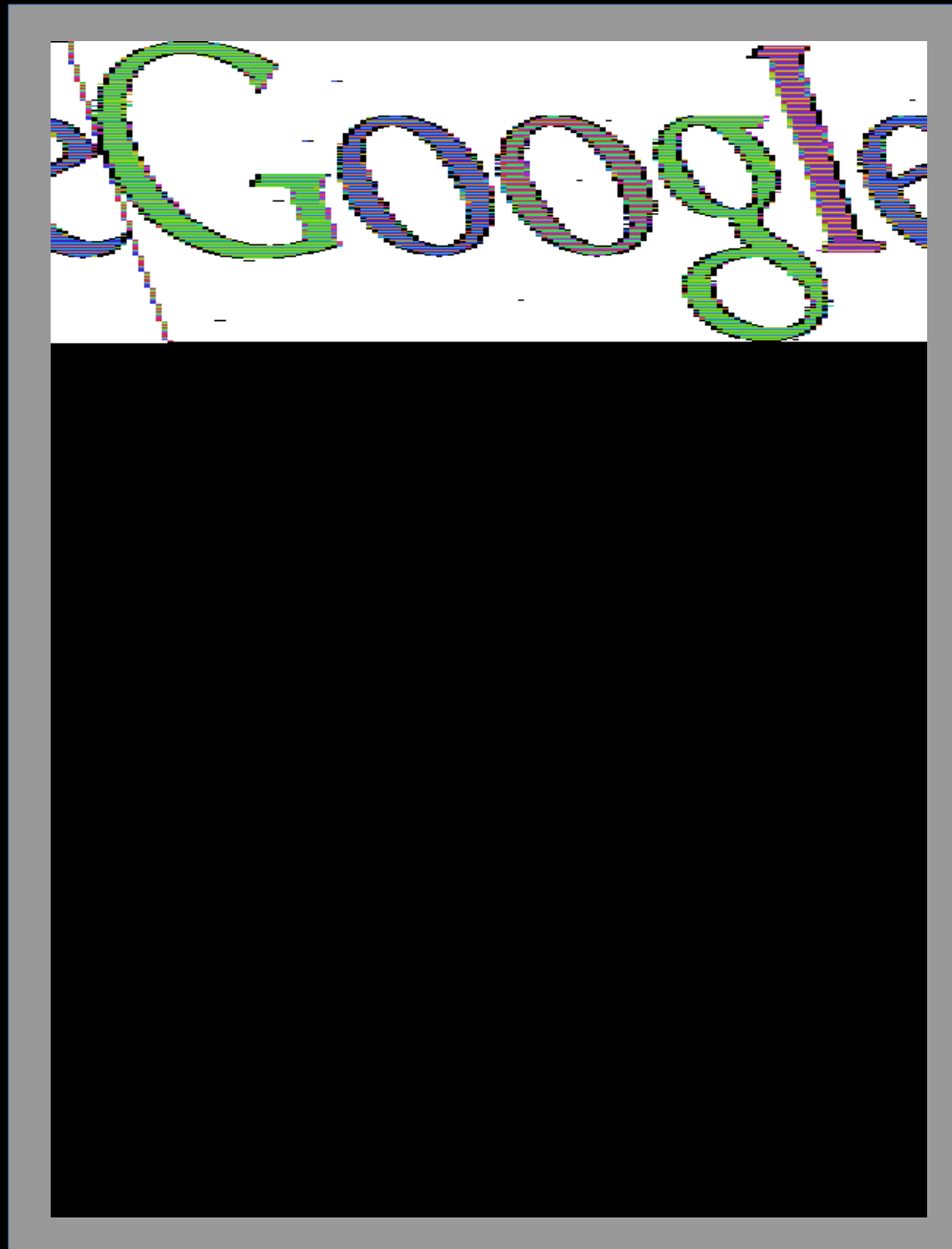
$$\text{ENC}(\text{[noise] Duck}) = \text{Google [noise]}$$

$$\text{DEC}(\text{Google [noise]}) = \text{Duck}$$

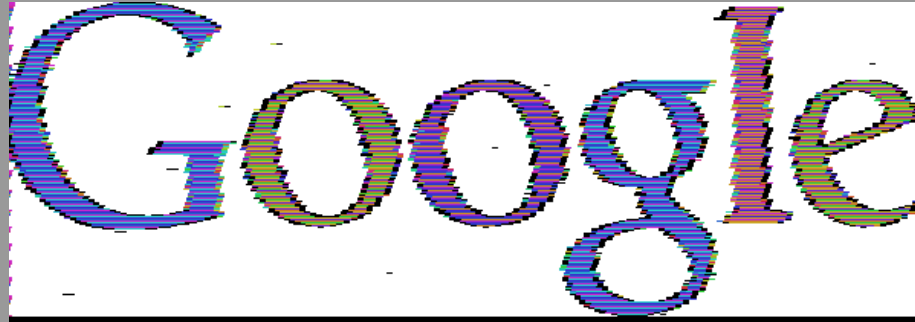


ENC  = 

```
$ ElectronicColoringBook.py encrypted.png  
-p4 -c255
```



```
$ ElectronicColoringBook.py combined.png  
-p4 -c255 -o3 -x 600.345
```

The image shows the Google logo in its characteristic multi-colored font (blue, red, yellow, green, blue). The logo is positioned at the top of a rectangular frame. The rest of the frame is a solid black rectangle.

plaintext

CBC
encrypted

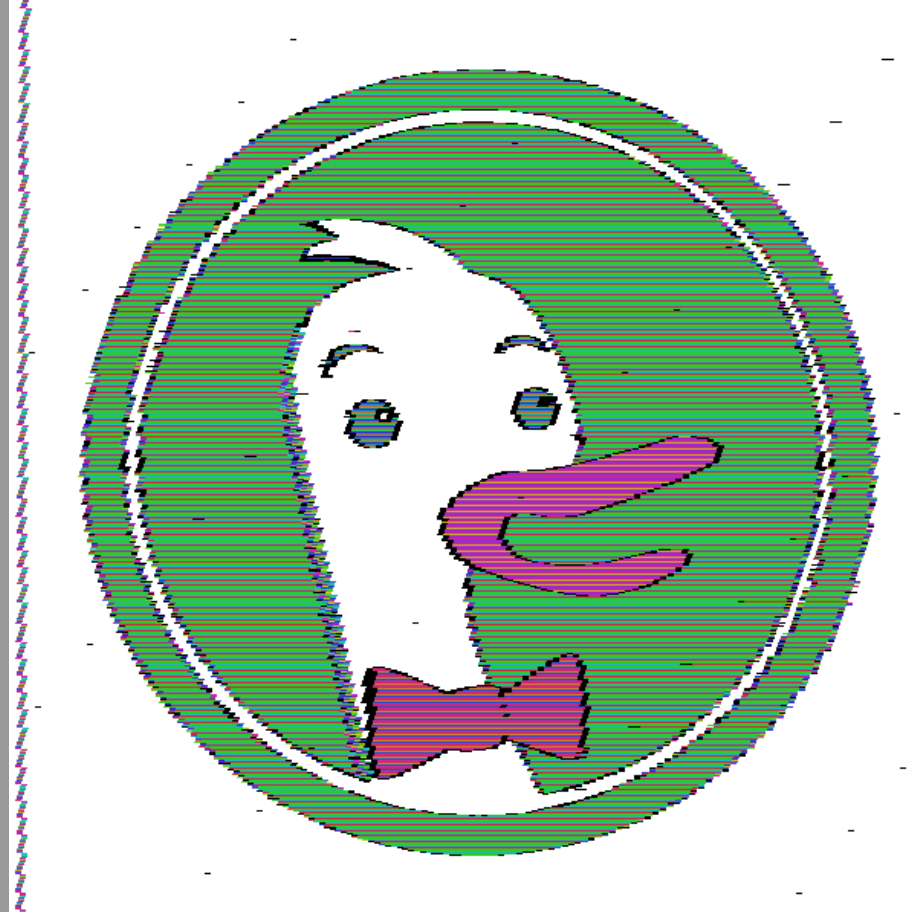
no repetition
= black

$$\text{DEC} \left(\begin{array}{c} \text{Google} \\ \text{[noise]} \end{array} \right) = \begin{array}{c} \text{[noise]} \\ \text{[DuckDuckGo logo]} \end{array}$$


```
$ ElectronicColoringBook.py decrypted.png  
-p4 -c255 -o3 -x 600.345
```

Google

¿CBC ?!

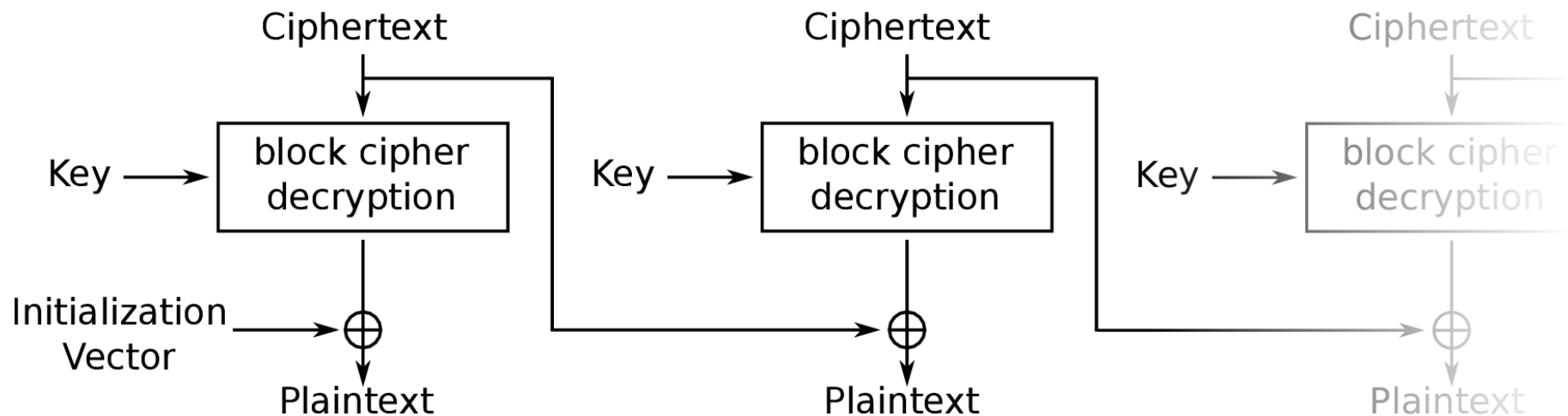


plaintext

How comes?

DEC(Google) = 

CBC decryption mode:



More in
POC || GTFO 0x06

[illegible]