!pip install pycrypto

Collecting pycrypto

Downloading pycrypto-2.6.1.tar.gz (446 kB)

|████████████████████████████████| 446 kB 5.1 MB/s

Building wheels for collected packages: pycrypto

Building wheel for pycrypto (setup.py) ... done

Created wheel for pycrypto: filename=pycrypto-2.6.1-cp37-cp37m-linux\_x86\_64.whl size=499932 sha256=e5084dbe99d19b87f78dc27ed3f75dcdb3c83532b3fd27eca54683c2538605d3

Stored in directory: /root/.cache/pip/wheels/cf/85/ba/bbd7c96add459de7598fb424e5ff2309baf2095c844ac0f191

Successfully built pycrypto

Installing collected packages: pycrypto

Successfully installed pycrypto-2.6.1

from Crypto.Cipher import AES

import string

import numpy as np

Длина секретного ключа K и соли S

SIZE\_KEY = 16

LETTERS = string.ascii\_uppercase

def generate\_string(size):

return ''.join(np.random.choice(list(LETTERS),size))

K = generate\_string(SIZE\_KEY).encode('ascii')

S = generate\_string(SIZE\_KEY)

print(f'Secret key: {K}\n salt: {S}')

def f(x):

if len(x) < len(S):

dif = len(S) - len(x)

x += S[:dif]

cipher = AES.new(K, AES.MODE\_ECB)

return cipher.encrypt(x.encode('ascii'))

pred\_salt = ''

while len(pred\_salt) < SIZE\_KEY:

x = ('A' \* SIZE\_KEY )[:-1 - len(pred\_salt)]

f\_without\_salt = f(x)

for letter in LETTERS:

if f(x + pred\_salt + letter) == f\_without\_salt:

pred\_salt += letter

break

assert pred\_salt == S

print(f'Predicted salt: {pred\_salt}')