Trabalho prático 1 - Estruturas Criptográficas

Autores: Ariana Lousada (PG47034), Cláudio Moreira (PG47844)

Grupo 12

Problema 1

Para a implementação da resolução deste exercício a equipa de trabalho tentou seguir uma estratégia de cifra “lightweight”.

Começou-se por definir o tamanho do bloco. Escolheu-se o tamanho 256 uma vez que se vai utilizar a cifra por blocos primitiva AES-256. Isto também porque um tamanho demasiado pequeno ou demasiado elevado poderá influenciar negativamente a segurança deste modo: por um lado pode facilitar o atacante, podendo este facilmente adivinhar as combinações de bits; por outro lado a cifra fica demasiado ineficiente para ser utilizada.

Neste problema é então inserida uma mensagem, a qual é dividida por blocos e posteriormente cifrada utilizando o AES-256.

Problemas de implementação

Apesar da equipa de trabalho ter iniciado o desenvolvimento de métodos de construção e aplicação do tweak, a sua implementação não foi alcançada.

Tal como mencionado anteriormente, o objetivo da equipa de trabalho seria usar uma estratégia de cifra "lightweight", na qual seria desenvolvido um tweak concatenado com a chave de longa duração. Este tweak iria ser constituído por um nounce único, ocupando BLOCKLEN/2 do seu tamanho total e pelo tamanho da mensagem total pretendida.

In [24]:

```
import os
from cryptography.hazmat.primitives.ciphers.aead import AESGCM
from binascii import hexlify as hexa
from os import urandom
BLOCKLEN = 256
def blocks(data):
    split = [hexa(data[i:i+BLOCKLEN]) for i in range(0, len(data), BLOCKLEN)]
    return b' '.join(split)


nonce = os.urandom(16)
message = b'\x00'*BLOCKLEN
message_authenticated = b'\x00'
key = AESGCM.generate_key(bit_length=256)
print ("key = %s\n" % k)

aes = AESGCM(key)
c = aes.encrypt(nonce,message,message_authenticated)

print("enc(%s) = %s \n" % (blocks(message), blocks(c)))

d = aes.decrypt(nonce,c,message_authenticated)

print("dec(%s) = %s \n" % (blocks(c), blocks(d)))


#nounceList = []
#b = 256 #tamanho de cada bloco

#def generate_tweak():
    #gerar o nonce
    #nonce = nounceGenerator

    #return nonce+(128).tobytes(128,'big')


# função que gera nounces únicos
#def nounceGenerator():
#    nounce = os.urandom(128) #b/2 = 128
#    if not (nounce in nounceList):
#        nounceList.append(nounce)
#        return nounce
#    else:
#        nounceGenerator()
```

```
key = b"\x92\xf8R\x80\x84|\x01,5\xbf'0\xdc\x10$\x9f"

enc(b'0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

```
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000') = b'4a4a5f037ed7754cdbd945a67519d587fb8f9ed4
99e90dd60f6258ba50dc9897c595d554f2e2c08fa25e4d15ccbaa8616820897572fc398a82f7e5b2f9635f8e5a5bfc823309680c68de2e97d
8d295c49eda2d667c05c0633df36da4612a5141e1ab603c5bfef29e807b39509438f38c68e34c52d7acca00b427d06764cf96ad2f16140f71
4ea89953515a0d4c76c4bdef626fafca34faa4fe560017e15c30f724bd40b9cfeb5fb1df200d30b14f351b1439fc39c23901bb310a1344dab
c48f9ededb1c80fa08829a3ec2aa83c01d7165c58ff8f499533ddfc2f9bb218cb2ed3608b90b18a6f74d763d83f9cf84a695eed00e1ed6fa4
6682001ece2601875098 dd0cca0799f337845fd6ddb36165918a'

dec(b'4a4a5f037ed7754cdbd945a67519d587fb8f9ed499e90dd60f6258ba50dc9897c595d554f2e2c08fa25e4d15ccbaa8616820897572f
c398a82f7e5b2f9635f8e5a5bfc823309680c68de2e97d8d295c49eda2d667c05c0633df36da4612a5141e1ab603c5bfef29e807b39509438
f38c68e34c52d7acca00b427d06764cf96ad2f16140f714ea89953515a0d4c76c4bdef626fafca34faa4fe560017e15c30f724bd40b9cfeb5
fb1df200d30b14f351b1439fc39c23901bb310a1344dabc48f9ededb1c80fa08829a3ec2aa83c01d7165c58ff8f499533ddfc2f9bb218cb2e
d3608b90b18a6f74d763d83f9cf84a695eed00e1ed6fa46682001ece2601875098 dd0cca0799f337845fd6ddb36165918a') = b'0000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000'
```