Tutorial #3

João Freitas up202100373
Rui Gonçalves up202103077

O presente relatório tem como objetivo descrever o processo seguido para a resolução da ficha `Tutorial #3`, disponibilizada no âmbito da disciplina de Criptografia Aplicada. As seções numeradas em baixo representam cada um dos exercícios resolvidos.

Modos de operação Cipher-Block Chaining e Counter

1) Utilize o Python para cifrar um ficheiro usando o modo de operação CBC

Usando a biblioteca `cryptography` do Python, facilmente implementa-se a cifração de uma mensagem usando a cifra de bloco AES junto com o modo de operação CBC:

```
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes

def encrypt_aes_cbc(input: bytes, key: bytes, iv: bytes):
    encryptor = Cipher(algorithms.AES(key), modes.CBC(iv)).encryptor()
    update = encryptor.update(input)
    finish = encryptor.finalize()
    return update + finish

def pad(m):
    return m+(chr(16-len(m) % 16)*(16-len(m) % 16)).encode()

key = os.urandom(16)
iv = os.urandom(16)

Path('input-key').write_bytes(key)
Path('input-iv').write_bytes(iv)

input = Path(sys.argv[1]).read_bytes()
cipher_input = encrypt_aes_cbc(pad(input), key, iv)
output = Path(sys.argv[2]).write_bytes(cipher_input)
```

2) Utilize o OpenSSL para comprovar que o ficheiro foi corretamente cifrado, ao decifrar o mesmo

Recorrendo ao seguinte comando, podemos indicar o OpenSSL a produzir o conteudo do ficheiro original, recorrendo ao ficheiro cifrado e a chave e iv usados na cifração.

```
openssl aes-128-cbc -d -in <cipher_input> -K <key_hex> -iv <iv_hex>
```

3) Atualize o valor de um byte do ficheiro e tenta decifrar de novo

Utilizando a ferramenta `Hex Fiend` para macOS, foi atualizado o valor de um byte num bloco aleatório. Ao correr o comando OpenSSL previamente descrito, observa-se que o ficheiro original foi decifrado, com algumas mensagens modificadas. Isto deve-se ao facto de se ter alterado o valor do byte.

> Seria possível recuperar o ficheiro cifrado com o modo CBC, se o IV e primeiro bloco tivessem corrumpidos/perdidos?

É possível recuperar partes do ficheiro sem o valor IV correto, pois este apenas é usado na operação XOR do primeiro bloco. Contudo, se o primeiro bloco fosse perdido ou corrumpido, não é possível decifrar o ficheiro, pois para decifrar um bloco, é necessário o anterior.

> Seria possível recuperar o ficheiro cifrado com o modo CBC, na ausência de um bit?

Desde que o bit ausente não influencie o primeiro bloco, é possível recuperar partes do ficheiro.

> Seria possível modificar um byte no ficheiro cifrado com o modo CBC, sem o recifrar por completo?

Desde que o byte modificado não influencie o primeiro bloco, é possível recuperar partes do ficheiro, logo este não iria recifrar o mesmo por completo.

4) Repita o exercício anterior com o modo CTR. Quais são as diferenças?

Para auxiliar a cifração do ficheiro com o modo CTR, foi utilizado o seguinte código Python:

```
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes

def encrypt_aes_ctr(input: bytes, key: bytes, nonce: bytes):
    encryptor = Cipher(algorithms.AES(key), modes.CTR(nonce)).encryptor()
    update = encryptor.update(input)
    finish = encryptor.finalize()
    return update + finish

def pad(m):
    return m+(chr(16-len(m) % 16)*(16-len(m) % 16)).encode()

key = os.urandom(16)
nonce = os.urandom(16)

Path('input-key').write_bytes(hexlify(key))
Path('input-nonce').write_bytes(hexlify(nonce))

input = Path(sys.argv[1]).read_bytes()
cipher_input = encrypt_aes_ctr(pad(input), key, nonce)
output = Path(sys.argv[2]).write_bytes(cipher_input)
```

Também, o comando OpenSSL previamente usado foi atualizado para corresponder ao modo CTR:

```
openssl aes-128-ctr -d -in <cipher_input> -K <key_hex> -iv <nonce_hex>
```

Atualizando um byte e decifrando, é possível recuperar o ficheiro original. Isto deve-se ao facto que no modo de operação CTR, diferentes blocos podem ser atualizados e posteriorment recuperar a mensagem, desde que o valor do contador (nonce) não seja perdido/corrumpido. Desta forma, e em contraste com o CBC, é possível recuperar o ficheiro se primeiro bloco tiver sido corrumpido/perdido.

Perdas de bits implicam que partes do ficheiro não sejam recuperadas. É também possível atualizar um byte aleatório no ficheiro, sem que este cifre de novo o conteúdo do ficheiro original.