

Zadanie 2


I-UPB

7. października 2021

Voľba riešenia zadania

Riešenie je naprogramované v Pythone a API použité pri šifrovaní sú z [pycrypto](#).

Spustenie aplikácie

Pred spustením programu je potrebné doinštalovať knižnice z **requirements.txt**. Riešenie neobsahuje žiadne GUI len CLI. 

Inštalácia

Inštalácia je nasledovná:

```
git clone https://github.com/behindbone/upb-02.git
pip install -r requirements.txt
```

Možnosti spustenia

```
python3 app.py -h
usage: app.py [-h] [--encrypt ENCRYPT] [--decrypt DECRYPT]

optional arguments:
  -h, --help            show this help message and exit
  --encrypt ENCRYPT      moznost na AES sifrovanie
  --decrypt DECRYPT      moznost na AES desifrovanie
```

Repozitár obsahuje aj priečinok **files** v ktorom sú súbory na odskúšanie šifrovania/dešifrovania.

Proces šifrovania

Spustíme nasledovne:

```
python3 app.py --encrypt [FILE]
```

Rovnako ako kľúč (*key*) a inicializačný vektor (*iv*) sú vytvorené náhoda a to vo výpise 1 na riadku 2 a 8. Následne sú obe uložené do textového súboru. Pri šifrovaní používam odporúčanú veľkosť bloku (*AES.block_size*), ktorá má prednastavenú hodnotu 16 bajtov. Metóda šifrovania je **CFB**, čiže šifruje každý blok nezávisle pomocou náhodného *iv*, ktorý je závislý od predošlého bloku.

```
1 def encrypt(file_name):
2     key = Random.new().read(AES.block_size)
3     write_key(key)
4     with open(file_name, 'rb') as f:
5         data = f.read()
6         cipher = AES.new(key, AES.MODE_CFB)
7         ciphertext = cipher.encrypt(pad(data, AES.block_size))
8         iv = b64encode(cipher.iv).decode('UTF-8')
9         ciphertext = b64encode(ciphertext).decode('UTF-8')
10    f.close()
11    with open(file_name + '.enc', 'w') as raw_data:
12        raw_data.write(iv + ciphertext)
13    raw_data.close()
```

Listing 1: Funkcia na AES šifrovanie súboru

Proces dešifrovania

Spustíme nasledovne (vstupom je zašifrovaný súbor):

```
python3 app.py --decrypt [FILE]
```

Kľúč na dešifrovanie prečítame z vytvoreného súboru (riadok 2, ukážka 2). Prvých 24 bajtov zašifrovaného súboru zodpovedá *iv*, preložíme si ho do formy bajtov, aby sme mohli dešifrovať.

Poznámka: Na riadku 13 vo výpise 2 ukladám dešifrovaný súbor odstránením prípony '.enc'.

```
1 def decrypt(file_name):
2     key = read_key()
3     with open(file_name, 'rb') as f:
4         data = f.read()
5         length = len(data)
6         iv = data[:24]
7         iv = b64decode(iv)
8         ciphertext = data[24:length]
9         ciphertext = b64decode(ciphertext)
10        cipher = AES.new(key, AES.MODE_CFB, iv)
11        decrypted = cipher.decrypt(ciphertext)
12        decrypted = unpad(decrypted, AES.block_size)
13        with open('decrypted_' + file_name[:-4], 'wb') as decrypted_f:
14            decrypted_f.write(decrypted)
15        decrypted_f.close()
```

Listing 2: Funkcia na AES dešifrovanie súboru

Výsledky

	Veľkosť	Čas v sekundách
1	5 MB	0,382
2	50 MB	2,719
3	500 MB	27,017
4	1 GB	55,265

Tabuľka 1: Tabuľka rýchlosti šifrovania

	Veľkosť	Čas v sekundách
1	5 MB	0,388
2	50 MB	2,759
3	500 MB	27,639
4	1 GB	56,527

Tabuľka 2: Tabuľka rýchlosti dešifrovania

Zdroje

- <https://www.youtube.com/watch?v=F2av7TaVc5Q>
- <https://github.com/the-javapocalypse/Python-File-Encryptor>
- <https://www.dlitz.net/software/pycrypto/>