Izvještaj iz 3. laboratorijskih vježbi - /SRP/

Symmetric key cryptography

Na prošlim laboratorijskim vježbama smo za izazov imali dekriptirati ciphertext koji se nalazio unutar datoteke čije je ime bilo 256-bitni hash. Da saznamo koju datoteku skinuti, morali smo svatko zasebno u Pythonu, koristeći *Cryptography* package, hashirati svoje ime u formatu (prezime-ime) i dodati ".encrypted".

> 💡 S ovim smo demonstrirali uporabu kriptografskih hash funkcija. Ako znamo kako ide poruka, onda ponovno hashiranje te poruke se treba podudarati sa danim hashom i onda znamo da se poruka dostavila nepromijenjena.

Nakon toga, krenuli smo na izazov dekripcije. Znali smo da se koristio AES enkripcijski algoritam sa 128-bitnim ključem. Također smo znali da je entropija toga ključa jednaka 22 bita. Što znači da zapravo key-space nije **2^128** (3.4028 * 10^38) nego **2^22** (4 194 304) što je za **8.11 * 10^31** manji broj mogućih ključeva. Zbog te činjenice, mi smo zapravo u mogućnosti dekriptirati zadani ciphertext u realnom vremenu koristeći brute-force napad.

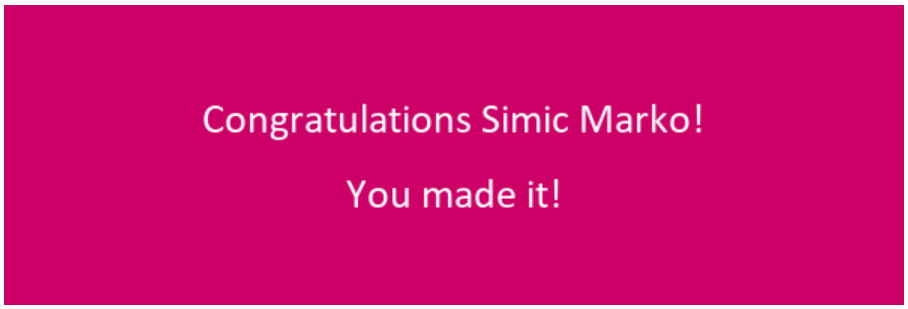
> 💡 S ovim smo demonstrirali da nije sve u količini bitova ključa s kojim se nešto enkriptira, nego koliko su nasumični ti bitovi (koliko se teško mogu predviditi).

Problem koji se treba riješiti je kako znati da je neki ključ valjan. Ključ je valjan ako dekriptirani ciphertext (plaintext) ima smisla. Ali, nema smisla da neki čovjek provjerava svaku kombinaciju ključa i ciphertexta koji će rezultirati nekim plaintextom.

Uz pomoć nekih hintova, zaključili smo da je plaintext koji tražimo, zapravo nije tekst (.txt) nego slika PNG formata. A znamo da bilo koji format datoteke sadrži neki metadata s kojim računalo koje mora prikazati taj format, zna s čime ima posla. PNG slike se lako prepoznaju jer kad se PNG slika transformira u tekst, prvih 8 byteova su "**\211PNG\r\n\032\n**". S ovim znanjem, možemo zaključiti da ćemo znati da je ključ ispravan, ako prvih 8 byteova dekriptiranog plaintexta započinju sa gore navedenih 8 byteova (ne mora nužno, ali jako su male šanse da sa neispravnim ključem dobijemo navedenih 8 byteova na početku plaintexta). S toga u petlju koju vrtimo za isprobavanje ključeva, stavimo provjeru, je li taj plaintext počinje s navedenih 8 byteova.

```
header = plaintext[:32]
            if test_png(header):
                print(f"Key found: {key}")
                with open("Bingo.png", "wb") as file3:
                    file3.write(plaintext)
                break
```

Nakon svega toga, ako smo sve uspješno odradili, dočekalo nas je iznenađenje.



Zaključak

Jako lako je sam sebe zavarati sa velikim brojevima, misleći da će te puno bitova zaštiti, ali to ne ide baš tako. Mogu povući paralelu sa načinom kako neki ljudi uče. Neki ljudi “uče” cijeli dan, a zapravo samo čitaju i kad prelistavaju gradivo nakon uspješnog dana učenja, govore sebi “ to znam, ovo znam, znam, znam, …”, a zapravo ništa nisu naučili. A neki ljudi uče 2 sata, tehnikom aktivnog ponavljanja, i tih 2 sata im mozak radi pri punom kapacitetu, i ti ljudi stvarno nauče to.

Šta želim reći, nije sve u kvantiteti, nego i u kvaliteti.