💻

Druga laboratorijska vježba

ZADATAK

Cilj ove vježbe je dobiti plaintext od odgovarajućeg ciphertexta u kontekstu simetrične kriptografije. Student nema pristup enkripcijskom ključu. Plaintext je enkriptiran pomoću sustava Fernet iz Pythonove biblioteke *cryptography*.

KREIRANJE VIRTUALNOG OKRUŽENJA

```
C:\Users\A507\filipfiric>python -m venv ffiric

C:\Users\A507\filipfiric>cd ffiric

C:\Users\A507\filipfiric\ffiric>cd Scripts

C:\Users\A507\filipfiric\ffiric\Scripts>activate

(ffiric) C:\Users\A507\filipfiric\ffiric\Scripts>

(ffiric) C:\Users\A507\filipfiric>pip install cryptography

(ffiric) C:\Users\A507\filipfiric>python
```

FERNET

```
from cryptography.fernet import Fernet

PLAINTEXT = b"Hello world"

key = Fernet.generate_key()

fernet = Fernet(key=key)

ciphertext = fernet.encrypt(PLAINTEXT)

deciphertext = fernet.decrypt(ciphertext)

print(f"{ciphertext}\n : \n{deciphertext}")
```

HASH-IRANJE IMENA

```
from cryptography.hazmat.primitives import hashes
import binascii

def hash(input):

     if not isinstance(input, bytes):
     input = input.encode()

     digest = hashes.Hash(hashes.SHA256())
     digest.update(input)
     hash = digest.finalize()
     return hash.hex()

filename = hash('prezime_ime') + ".encrypted"

if __name __ == "__main __":
      print(hash("firic_filip"))
```

PRONALAZAK ENKRIPCIJSKOG KLJUČA BRUTE FORCE-OM

```
import base64
from cryptography.hazmat.primitives import hashes
from cryptography.fernet import Fernet

def hash(input):
    if not isinstance(input, bytes):
        input = input.encode()
    digest = hashes.Hash(hashes.SHA256())
    digest.update(input)
    hash = digest.finalize()
    return hash.hex()

def test_png(header):
    if header.startswith(b"\211PNG\r\n\032\n"):
        return True

def brute_force():
    filename =
"a48de12cdad23f546768a64219baf69739eba3d52d0c5ca393b01fc57a624667.encrypted"
    with open(filename, "rb") as file:
        ciphertext = file.read()
    ctr = 0
    while True:
        key_bytes = ctr.to_bytes(32, "big")
        key = base64.urlsafe_b64encode(key_bytes)
        if not (ctr + 1) % 1000:
            print(f"[*] Keys tested: {ctr + 1:,}", end="\r")
        try:
            plaintext = Fernet(key).decrypt(ciphertext)
            header = plaintext[:32]
            if test_png(header):
                print(f"[+] KEY FOUND: {key}")
                with open("BINGO.png", "wb") as file:
                    file.write(plaintext)
                break
        except Exception:
            pass
        ctr += 1

if __name__ == "__main__":
    # hash_value = hash("firic_filip")
    # print(hash_value)
    brute_force()
```