Lab 3: Symmetric key cryptography

Zadatak je bio riješiti crypto izazov, na način da smo prvo trebali pronaći vlastitu datoteku na serveru te je preuzeti.Pri čemu su imena datoteka bile hash vrijednosti SHA-256 funkcije koja je kao argument uzimala naše ime i prezime.

Te smo ime saznali pomoću:

```
from cryptography.hazmat.primitives import hashes


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("botic_katarina") + ".encrypted"
```

Te nakon što smo saznali koja je naša datoteka potrebno ju je bilo dešifrirati i spremiti u određenu datoteku.

Za enkripciju je korišten **ključ ograničene entropije - 22 bita.**

Te smo saznali da je datoteka slika u png pa plaintext treba započinjati sa “\211PNG\r\n\032\n” (prvih 8 byteova karakterističnih za png format). Te kako bi dobili dešifriranu datoteku odlučili smo koristiti brute-force napad.Tako da smo počeli od ključa 0 te smo za svaki ključ provjeravali je li file koji dobijemo dešifriranjem png formata ukoliko nebi bio uvečali bi ključ za 1 i ponovno pokušali. Na kraju je rješenje bilo slika na kojoj je pisalo Congratulations Botic Katarina! You made it!

cijeli kod:

```
import base64
from os import path
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
    return False


def brute_force(ciphertext):
    ctr = 0
    while True:
        key_bytes = ctr.to_bytes(32, "big")
        key = base64.urlsafe_b64encode(key_bytes)

        # Now initialize the Fernet system with the given key
        # and try to decrypt your challenge.
        # Think, how do you know that the key tested is the correct key
        # (i.e., how do you break out of this infinite loop)?

        try:
            plaintext = Fernet(key).decrypt(ciphertext)
            header = plaintext[:32]

            if test_png(header):
                print(f"BINGO: {key}")
                with open("BINGO.png", "wb") as file:
                    file.write(plaintext)
                break
        except Exception:
            pass

        ctr += 1
        if not ctr % 1000:
            print(f"[*] Keys tested: {ctr:,}", end="\r")
```

```
if __name__ == "__main__":
    filename = hash("botic_katarina") + ".encrypted"

    # Create a file with the filename if it does not alread exist
    if not path.exists(filename):
        with open(filename, "wb") as file:
            file.write(b"")


#Open your challenge file and read in your challenge
with open(filename, "rb") as file:
    ciphertext = file.read()

# print(ciphertext)
# Start the attack
brute_force(ciphertext)
```