



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

**APLIKACE PRO STEGANOGRFII MÉDIA**

**SEMESTRÁLNÍ PROJEKT**

SEMESTRAL PROJECT

**AUTOŘI PRÁCE**

AUTHORS

Andrej Krivulčík, Pavla Hlučková, Matěj Miška, Jiří Březina

**BRNO 2021**

# Obsah

|          |                                  |           |
|----------|----------------------------------|-----------|
| <b>1</b> | <b>Úvod</b>                      | <b>3</b>  |
| <b>2</b> | <b>Návrh</b>                     | <b>4</b>  |
| <b>3</b> | <b>Teoretický rozbor</b>         | <b>5</b>  |
| <b>4</b> | <b>Řešení</b>                    | <b>5</b>  |
| <b>5</b> | <b>Aplikace</b>                  | <b>9</b>  |
| 5.1      | Používání aplikace . . . . .     | 9         |
| a)       | První sekce . . . . .            | 9         |
| b)       | Druhá sekce . . . . .            | 9         |
| c)       | Třetí sekce . . . . .            | 11        |
| <b>6</b> | <b>Dokumentace kódu aplikace</b> | <b>12</b> |
| a)       | SteganoImage.py . . . . .        | 12        |
| b)       | SteganoVideo.py . . . . .        | 12        |
| c)       | GUI.py . . . . .                 | 12        |
| <b>7</b> | <b>Závěr a zhodnocení práce</b>  | <b>13</b> |
| <b>8</b> | <b>Literatura</b>                | <b>14</b> |

# 1 Úvod

Tento projekt se zabývá tvorbou aplikace na využití steganografie média. Cílem práce je vytvoření samostatné aplikace pro úpravu média za účelem skrytí informace. Aplikace je vytvořena programovacím jazykem Python s použitím knihoven na úpravu multimediálních dat (`Pillow`, `ffmpeg`, `moviepy`) a k vytvoření grafického uživatelského rozhraní (`PyQt5`). Uživatel má možnost zvolit formát multimediálních dat a zprávu, která je v médiu skryta, společně s heslem zajišťujícím lepší zabezpečení přenesených dat proti úmyslnému poškození souboru a brute-force útokům na odhalení zprávy. Dále aplikace podporuje také zpětné dešifrování informace z média po zadání správného hesla. Již zmíněné formáty multimediálních dat podporovaných aplikací jsou formáty obrázku `.png`, `.jpg` a formát videa `.mp4`. Po zpracování se obrázky uloží jako `.png` bez alfa kanálu a videa jako `.mov`. Aplikace disponuje intuitivním grafickým rozhraním pro jednoduchou práci s médii.

## 2 Návrh

Cílem je vytvořit uživatelsky přívětivou aplikaci, která by umožňovala jednoduché skrytí zprávy do obrázku a videa, aniž by to bylo lidským okem rozeznatelné. Aby byla zpráva lépe zabezpečena, není kódována do obrázku pixel po pixelu, ale k určení zakódovaných pixelů je využit generátor náhodných čísel. Tento generátor má jako „seed“ vlastní heslo, které uživatel zadá. Toto stejné heslo musí být zadáno i při dekódování zprávy, jinak nebude zpráva čitelná.

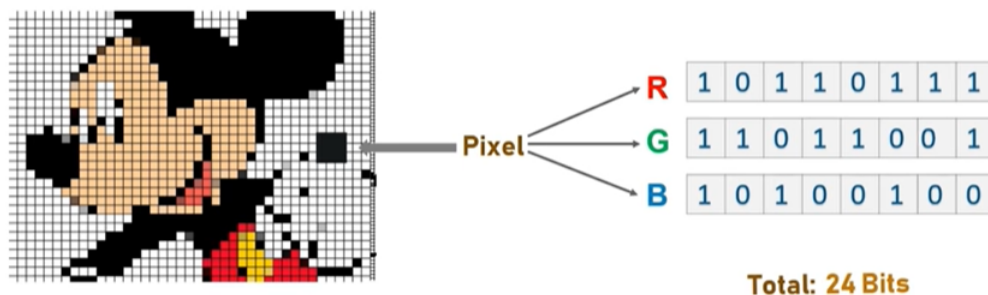
Co se týče kódování zprávy do videa, probíhá obdobným způsobem s malými obměnami. Video musí být před vložením zprávy rozděleno do snímků, které následně nesou zprávu. Samotná zpráva je v případě videa kódována po znacích pokaždé do jiného snímku, na jinou pozici. Video je po kódování opět sestaveno a dekódování zprávy není možné bez kódovacího hesla (používáno k odvození dalších parametrů) a opětovného rozdělení do snímků.

Aplikace obsahuje grafické uživatelské rozhraní, které je přehledně rozdělené do sekcí. Umožňuje tak bezproblémovou a jednoduchou obsluhu. Po načtení obrázku ke kódování je zobrazen jeho náhled, po ukrytí zprávy je zobrazen i náhled zakódovaného obrázku k porovnání.

### 3 Teoretický rozbor

Steganografie je vědní disciplína mající za úlohu přenos tajné zprávy v pozadí média skrze neutajenou komunikaci.

Podstatou této manipulace obrazu je skrytí zprávy do hodnot RGB pixelů v médiu. Pro lidské oko je tato změna nezaznamenanatelná, avšak při důkladném náhledu lze ze změny parity hodnot RGB vyčíst bitový řetězec obsahující zprávu.



Obr. 1: Princip rozložení hodnot RGB souřadnic.

V případě zvolení videa jako média pro uchování zprávy, je nutné dané video rozložit do samostatných snímků, z nichž je video složeno, a následně zprávu vložit zprávu do jednoho či více snímků. Tyto snímky je třeba následně znovu složit do vzoru původního videa.

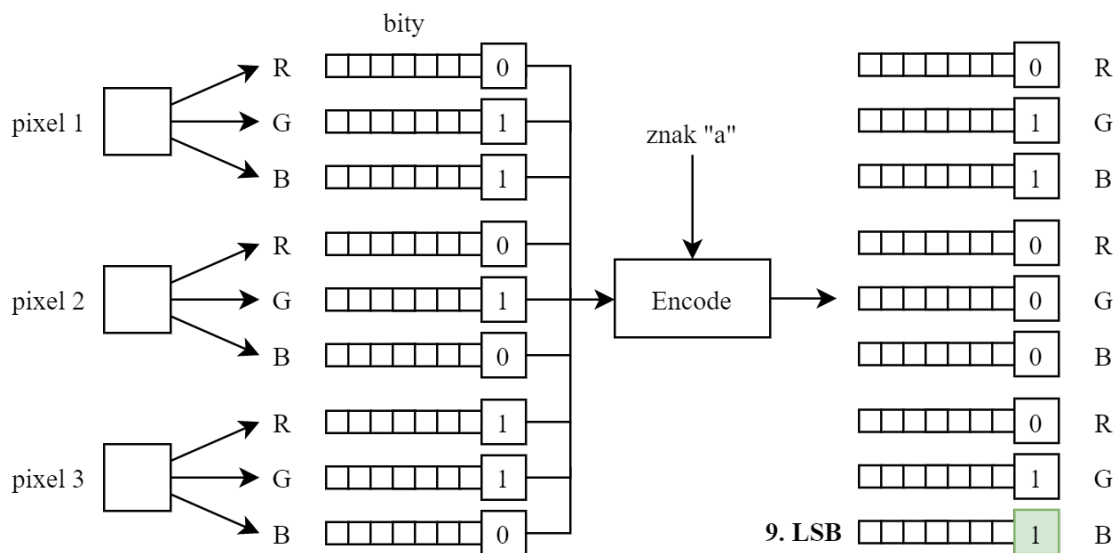
### 4 Řešení

Aplikace je vytvořena programovacím jazykem Python s použitím knihoven na úpravu multimediálních dat (Pillow, ffmpeg, moviepy) a k vytvoření grafického uživatelského rozhraní (PyQt5).

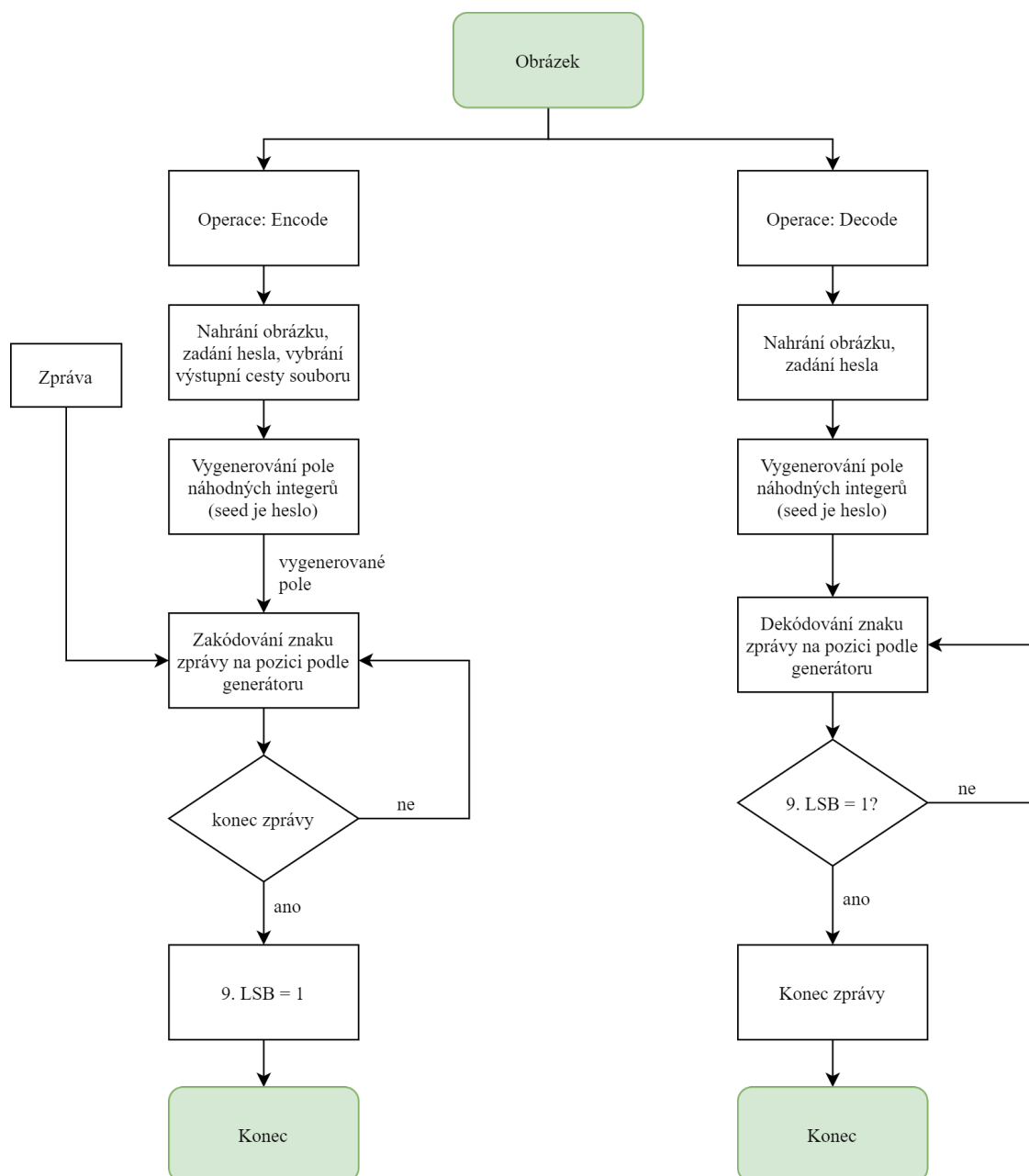
Proces kódování spočívá v tom, že pomocí knihovny Pillow se obrázek rozloží na pole hodnot RGB, které se následně upravují o hodnotu 1, a tedy dochází k úpravě hodnoty posledního bitu (Least Significant Bit). Pro lidské oko je tato změna prakticky neviditelná. Pomocí náhodného generátoru, beroucího jako svůj „seed“ heslo zadané uživatelem, se zpráva rozdělí s odsazením vytvořeným pomocí taktéž vygenerované délky. Takovýmto způsobem je dosaženo toho, že zpráva není koncentrována v jedné části obrázku, ale je plošně rozprostřena do celého obrázku. Tímto způsobem je nemožné bez znalosti hesla získat zprávu z obrázku. Výsledný formát obrázku je .png, jelikož při jeho vytváření nedochází ke kompresi. Vstupem může být libovolný soubor .png alebo .jpg.

Podobný proces probíhá při procesu dešifrování, kdy se pomocí náhodného generátoru (opět zde heslo funguje jako „seed“) generují odsazení mezi načítáním jednotlivých RGB složek.

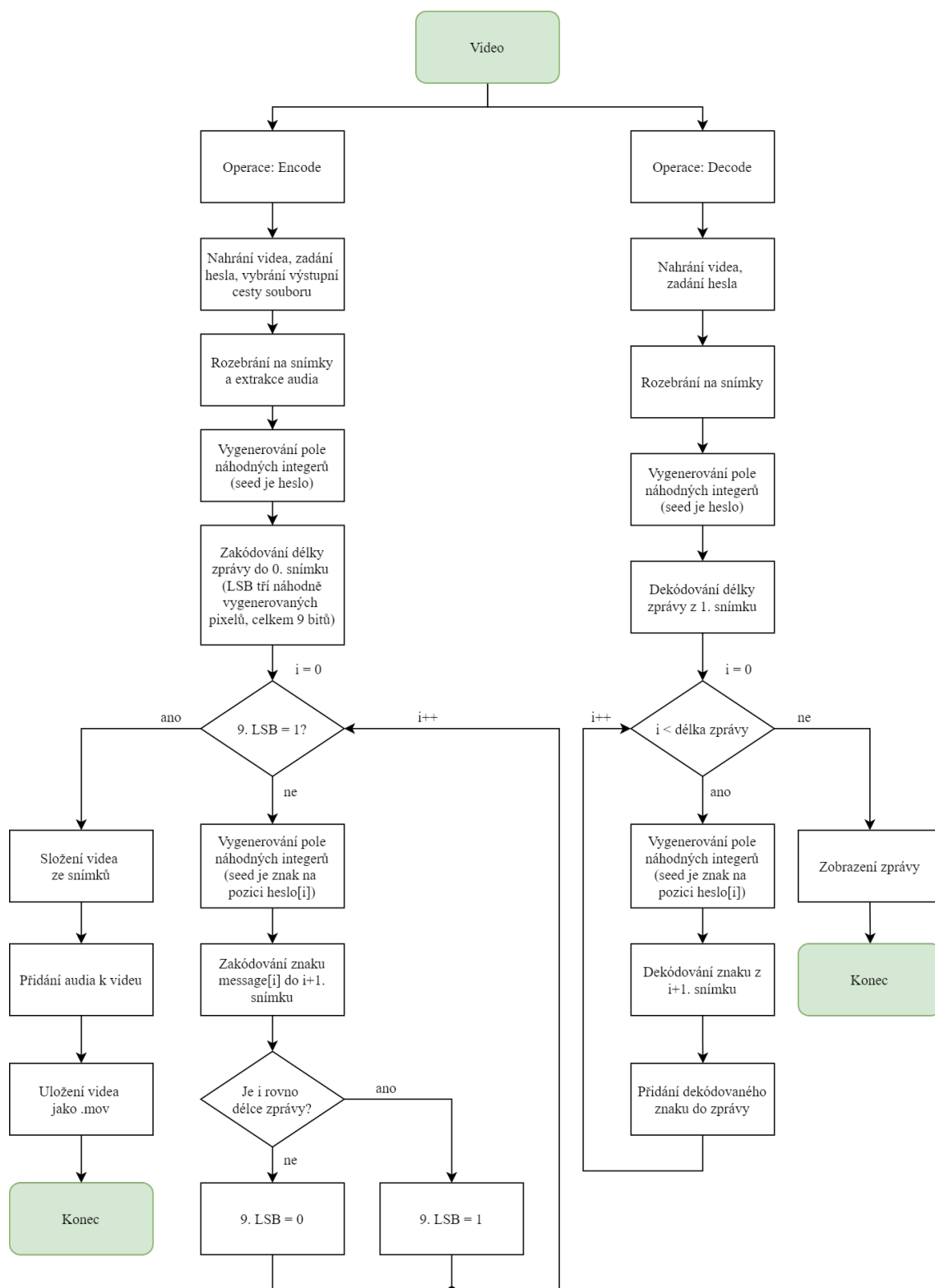
Proces šifrování a dešifrování videa spočívá v rozložení videa na jednotlivé snímky pomocí knihovny `ffmpeg`, knihovna `moviepy` vyextrahuje audio a následně aplikace rozloží zprávu na jednotlivé znaky postupně vkládané do snímků v původní posloupnosti (stejná, jako je ve zprávě). Aktuální vkládání zprávy do videa reprezentuje tzv. proof-of-concept, tedy že je možné vkládat zprávu do jednotlivých libovolných snímků videa. Následně je video opět složeno, tentokrát však již s upravenými snímky. Nevýhodou tohoto postupu je, že proces skládání videa nemůže obsahovat žádnou kompresi, tzn. video bude o mnoho větší velikost, než by po kompresi obvykle mělo. Takové video se uloží ve formátu `.mov`.



Obr. 2: Princip zakódování znaku do pixelů.



Obr. 3: Vývojový diagram kódování a dekódování zprávy v obrázku.

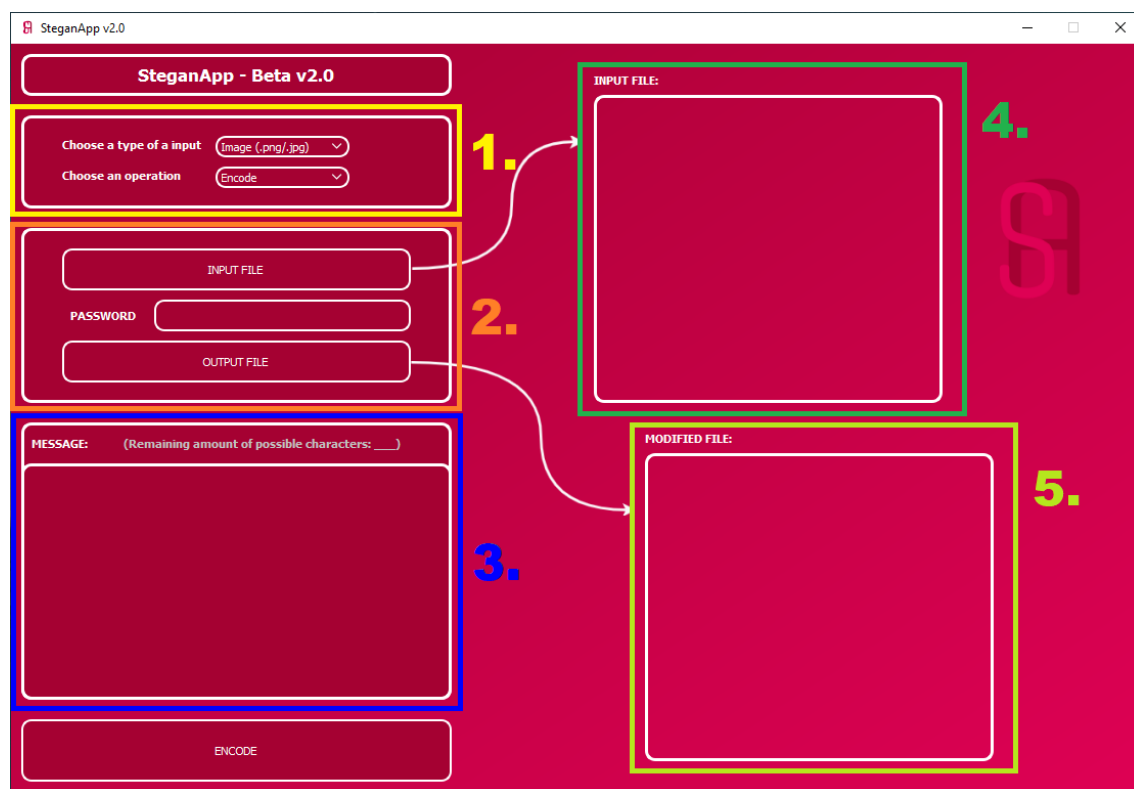


Obr. 4: Vývojový diagram kódování a dekódování zprávy ve videu.



## 5 Aplikace

Aplikace samotná je uživatelsky velmi přívětivá. Je přehledně rozdělena do několika sekcí, přičemž uživatelsky jsou přístupné první tři tyto sekce a ostatní už slouží pouze pro kontrolu viz obr. 5.



Obr. 5: Základní rozhraní aplikace se zvýrazněním sekcí.

### 5.1 Používání aplikace

#### a) První sekce

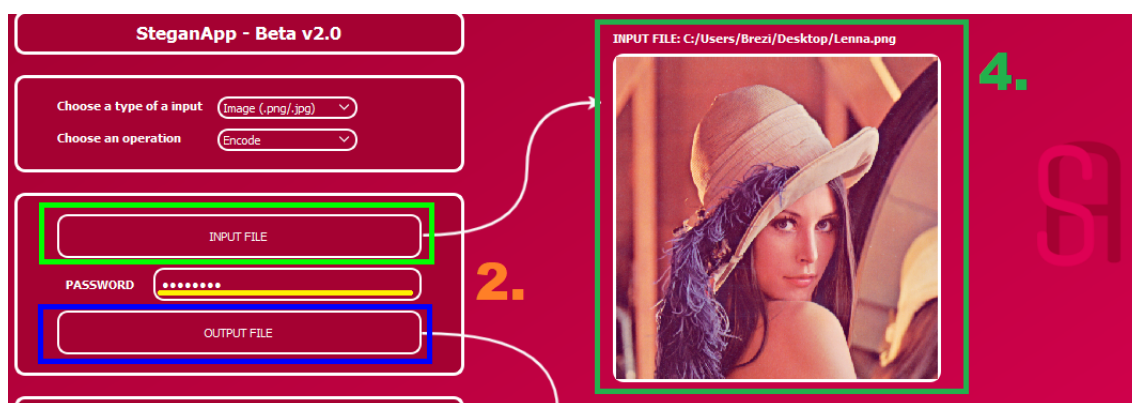
V **první sekci** vybere **typ souboru** *Image (.png/.jpg)* (obrázkové soubory ve formátu PNG nebo JPG) nebo *Video (.mp4)* (video soubory ve formátu MP4) a samotnou **operaci** *Encode* pro šifrování, případně *Decode* pakliže chce dešifrovat již vytvořený soubor.

#### b) Druhá sekce

Podle výběru operace (šifrování či dešifrování) v předchozí sekci se uživateli zobrazí dostupné možnosti ve **druhé** sekci. V případě, že uživatel zvolil operaci *Encode*, tak se mu zobrazí v této sekci celkem **dvě tlačítka** – **první tlačítko** pro výběr

zdrojového souboru (*INPUT FILE*), kdy se načtený obrázek zobrazí uživateli ve **čtvrté sekci** i včetně cesty k souboru. **Druhé tlačítko** slouží pro pojmenování a výběr umístění výsledného obrázku (*OUTPUT FILE*). Mezi tlačítky je prostor pro zadání **hesla** k šifrování souboru. Pro samotné šifrování je potřeba ještě napsat zprávu ve třetí sekci.

Pokud by uživatel zvolil v předchozí sekci operaci *Decode*, tak se mu v této sekci pochopitelně **nezobrazí** tlačítko *OUTPUT FILE* pro výsledný soubor, jelikož při dešifrování se již žádný další soubor nevytváří a pouze se zobrazí šifrovaná zpráva ve třetí sekci.



Obr. 6: Detail druhé a čtvrté sekce se zvýrazněním jednotlivých prvků.

### c) Třetí sekce

Ve **třetí sekci** uživatel zadá samotnou **zprávu** do pole *MESSAGE:*, kterou chce do obrázku zašifrovat. Nad polem zprávy je zobrazen **počet znaků**, který zbývá do maximálního počtu znaků, které je možné do zprávy zakomponovat. Po kliknutí na **tlačítko** *ENCODE* se provede samotné šifrování a následně se v poslední **páté sekci** zobrazí výsledný zašifrovaný obrázek.

V případě, že by uživatel v **první** sekci vybral operaci *Decode*, tak se v poli *MESSAGE:* zobrazí dešifrovaná zpráva z vybraného obrázku (samozřejmě pokud zadá správné heslo).



Obr. 7: Detail třetí a páté sekce se zvýrazněním jednotlivých prvků.

## 6 Dokumentace kódu aplikace

### a) `SteganoImage.py`

`convertDataFromString(...):`

překonvertování zprávy do binární podoby

`modifyPixels(...):`

zakódování zprávy do trojice pixelů

`encodeToImage(...):`

vytvoření, zakódování a uložení zprávy do obrázku

`decodeFromImage(...):`

dekódování zprávy z obrázku

### b) `SteganoVideo.py`

`audioFromVideo(...):`

vyjmutí audio-stopů z videa

`videoToFrames(...):`

překonvertování videa do samostatných snímků

`framesToVideo(...):`

překonvertování snímků společně s původní audio-stopou zpět do videa

`encodedVideoToFrame(...):`

překonvertování upraveného videa do samostatných snímků

`deleteFrames(...):`

vymazání nadbytečných souborů po dokončení zpětného složení videa

`videoEncode(...):`

kódování samotné zprávy do videa

`videoDecode(...):`

dekódování zprávy z videa

### c) `GUI.py`

obsahuje obslužné metody ke grafickému rozhraní a jeho nastavení

## 7 Závěr a zhodnocení práce

Aplikace je funkční a disponuje příjemným uživatelským rozhraním. Funguje pro formáty obrázku JPG a PNG a pro videa ve formátu MP4. Ke zprovoznění aplikace je nutné nainstalovat `ffmpeg` pro příkazovou řádku Windows a přidat jej do systémové proměnné `PATH`, nebo dané EXE soubory přesunout přímo do složky `System32`. Dále je nutné nainstalovat již zmíněné knihovny `Pillow`, `moviepy`, `PyQt5` a `ffmpeg-python`.

## 8 Literatura

- [1] *Steganography*. [online]. [cit. 2021-04-26]. Dostupné z: <https://en.wikipedia.org/wiki/Steganography>
- [2] *Steganography in video*. [online]. [cit. 2021-04-26]. Dostupné z: <https://bit.ly/32OGkG9>
- [3] *Python-ffmpeg*. [online]. [cit. 2021-04-26]. Dostupné z: <https://kkroening.github.io/ffmpeg-python/>
- [4] *MoviePy*. [online]. [cit. 2021-04-26]. Dostupné z: <https://zulko.github.io/moviepy/>
- [5] *Pillow*. [online]. [cit. 2021-04-26]. Dostupné z: <https://pillow.readthedocs.io/en/stable/>
- [6] *PyQt5*. [online]. [cit. 2021-04-26]. Dostupné z: <https://pypi.org/project/PyQt5/>
- [7] *Qt Designer Manual*. [online]. [cit. 2021-04-26]. Dostupné z: <https://doc.qt.io/qt-5/qtdesigner-manual.html>
- [8] *RGB distribution*. [online]. [cit. 2021-04-26]. Dostupné z: <https://bit.ly/3aFssSR>