Semestrální práce z KIV/BIT

Steganografie libovolného textu v souboru formátu AVI

Vladimír Láznička
5. 5. 2016

Odhadovaná doba řešení – 17 hodin

Obsah

Popis problému....3
Analýza problému – rozbor AVI souboru....3
RIFF hlavička....3
Seznamy a datové kousky....4
Seznamy 'hdrl' a 'movi'....4
Využití pro steganografii....5
Návrh řešení....6
Popis řešení....6
Ukrytí zprávy do souboru....6
Nalezení zprávy v souboru....7
Uživatelská dokumentace....8
Závěr....9

Popis problému

Cílem práce je vyzkoušení možnosti **skrývat libovolný text pomocí vybrané steganografické metody do souboru formátu *AVI***.

Řešení problému by mělo zahrnovat tvorbu programu schopného načíst typický AVI soubor a pomocí steganografie v jeho obsahu ukrýt předanou textovou zprávu, **aniž by se při jeho přehrávání projevily nějaké člověkem pozorovatelné změny**, které by mohly poukazovat na přítomnost zprávy.

Program by měl poté rovněž umožnit skrytou zprávu z takto upraveného souboru zase načíst a zobrazit v člověku jasně čitelné podobě.

Analýza problému – rozbor AVI souboru

Soubor ve formátu *AVI* (Audio Video Interleave) odpovídá takzvané specifikaci *RIFF* (Resource Interchange File Format), která určuje jeho celkovou strukturu. Obecně vzato může soubor formátu AVI obsahovat více datových proudů s rozdílnými typy dat. **Typicky obsahují AVI soubory proudy obrazových a zvukových dat**, které jsou pak čtené přehrávači.

Soubory AVI se mohou vyskytovat dvojího typu – Původní *AVI 1.0* specifikované firmou *Microsoft* a pozdější formáty *OpenDML AVI* (viz článek - [AVI RIFF File Reference]). **Tato práce se zaobírá pouze původním formátem**, tudíž nepočítá s možnostmi otevřených formátů a není zaručené, že výsledné řešení pro ně bude fungovat (ale je to možné).

RIFF hlavička

Soubor podléhající specifikaci RIFF pak vždy obsahuje *RIFF hlavičku*, která je následována volitelným počtem *seznamů* (*lists*) a *datových kousků* (*chunks*), teoreticky i nulovým. Hlavička má následující podobu:

```
RIFF <fileSize> <fileType>
```

kde '`RIFF`' je **4-bytový identifikátor** souboru specifikace RIFF, *fileSize* je pak velikost celého souboru v bytech (zaznamenaná jako **32-bitový DWORD**), počínaje **od bytu za touto hodnotou**, a nakonec *fileType*, který označuje typ souboru (v takzvaném **FOURCC datovém typu**, což je technicky **32-bitový kladný integer**, který je složen ze 4 po sobě jdoucích znaků) **v případě AVI souborů** se jedná o identifikátor '`AVI `' (vč. mezery za).

Seznamy a datové kousky

Seznamy (*lists*) i *datové kousky* (*chunks*) mají svou vlastní strukturu, která se do určité míry podobá. Seznamy mají podobu následující:

```
LIST <listSize> <listType> <listData>
```

kde '`LIST`' je **4-bytový literál** naznačující, že se jedná o seznam, *listSize* je potom velikost seznamu v bytech **zahrnující kromě dat i hodnotu *listType***, který klasicky identifikuje, o jaký seznam se jedná. Následují pak samotná data seznamu o délce 0 až N bytů. Tato data potom mohou obsahovat i další seznamy nebo datové kousky.

Podoba datových kousků je následující:

```
<ckID> <ckSize> <ckData>
```

kde *ckID* je **4-bytový identifikátor** (datový typ FOURCC) toho, jaká data budou v tomto kousku obsažena. Velikost dat *ckSize* je opět **32-bitová hodnota** udávající velikost následovaných dat (*ckData*) v bytech.

Seznamy 'hdrl' a 'movi'

Soubor AVI musí povinně definovat následující **dva typy seznamu** – '`hdrl`', který představuje hlavičku AVI souboru a je logicky umístěn ihned za hlavičku RIFF, a '`movi`', který představuje typicky **audio nebo video datový proud rozdělený do jednotlivých datových kousků**.

Seznam '`hdrl`' obsahuje několik dalších listů s informacemi o AVI souboru – '`strl`', '`strh`' (hlavička datového proudu) a '`strf`' (formát datového proudu).

Seznam '`movi`' je pak složen z datových kousků **popsaných 4-bytovým identifikátorem**

`XXaa`, kde '`XX`' značí číslo datového proudu (00, 01, 02…) a '`aa`' je pak dvouznakový kód definující typ dat konkrétního kousku, kde je:

- `db` – nekomprimovaný video snímek
- `dc` – komprimovaný video snímek
- `pc` – změna barevné palety
- `wb` – audio data

Využití pro steganografii

Vzhledem k výše popsaným strukturám AVI souboru bude **zřejmě nejpřijatelnější soustředit se na skrývání zprávy do datových kousků v seznamu '`movi`'**, neboť ty nám poskytnou nejvíce prostoru , a to způsobem, který by neměl viditelně ovlivnit podobu nebo kvalitu videa/zvuku v datovém proudu.

Návrh řešení

Pro demonstraci steganografie textu v AVI souboru bude využita jedna z nejjednodušších metod – ***Least Significant Bit*** (LSB) nebo-li **metoda nejméně významového (nebo také posledního při vhodném zakončování bytů) bitu**. Jednoduše řečeno – zpráva se bude „rozebírat“ bit po bitu a tyto bity se budou vkládat na místo nejméně významových bitů v jednotlivých bytech souboru.

Jako místo souboru, kde bude k ukrývání zprávy docházet, bude nejvhodnější použít **datové byty umístěné v datových kouscích seznamu** '`movi`'. Aby bylo možno ukládat i velmi dlouhé zprávy, bude zapotřebí vyřešit situaci, kdy počet bitů zprávy přesáhne počet bytů jednoho datového kousku. Zprávu tedy bude vhodné **rozdělit způsobem, aby první znak, který se již celý nevejde do daného kousku, byl použit jako první znak ve zbytku zprávy, která se začne ukládat do dalšího kousku**. Tento princip bude samozřejmě třeba dodržet i při čtení ukryté zprávy.

Zpráva ukrývaná do dat AVI souboru by rovněž měla být vybavena nějakou **jednoznačnou ukončovací sekvencí**, aby poté při jejím získávání mohl program skládání výstupu ukončit, jakmile ji celou najde.

Popis řešení

Řešením problému je program, který pracuje ve dvou možných, navzájem výlučných, režimech – **schování předané zprávy do vybraného souboru** (resp. vytvoření kopie souboru s vloženou zprávou na vybraném místě) a **nalezení ukryté zprávy ve vybraném souboru**.

Pozn.: vzhledem k omezením velikosti klasických polí ve vybraném programovacím jazyku a implementovaném způsobu zpracování bytů, musí být soubor **menší než 2GB**.

Ukrytí zprávy do souboru

Vybraný vstupní soubor je **načítán do pole bytů**, které pak drží obsah souboru pro další práci. Jakmile je k dispozici toto pole, je postupným procházením jednotlivých bytů **nalezena znaková sekvence '`movi`'**, která značí seznam obsahující datový proud AVI souboru.

Zpráva (s přidanou ukončovací sekvencí) se následně předá metodě, která ji bude ukládat po bitech do jednotlivých datových kousků. Před vstupem na konkrétní kousek se provede kontrola, **jestli má nenulovou velikost**. Pokud je tato podmínka splněna jednotlivé znaky zprávy jsou

pomocí bitových operací uloženy po **bitech do jednotlivých bytů v poli metodou *LSB***.

Pokud je zpráva delší, než by bylo třeba, aby se vešla do jednoho datového kousku, je zbývající část zprávy **vrácena z metody a použita jako vstup do této metody v dalším průběhu cyklu**.

Jakmile je celá zpráva uložena do pole s byty, je toto pole **zapsáno na disk na vybranou pozici**.

Nalezení zprávy v souboru

Vstupní soubor je načten do pole bytů stejně jako v předchozím případě a opět je nalezena **znaková sekvence `'movi'`** značící seznam s hledaným datovým proudem. Zpráva je poté skládána z nejméně významových bitů v jednotlivých bytech datových kousků.

Celý proces je ukončen v momentě, kdy zpráva na svém konci **obsahuje programem pevně specifikovanou ukončovací sekvenci**. Pokud není taková sekvence nalezena, jako zpráva se vrátí **bity získané ze všech datových kousků** s nenulovou velikostí.

Jakmile je zpráva celá získána, je vypsaná na výstup programu.

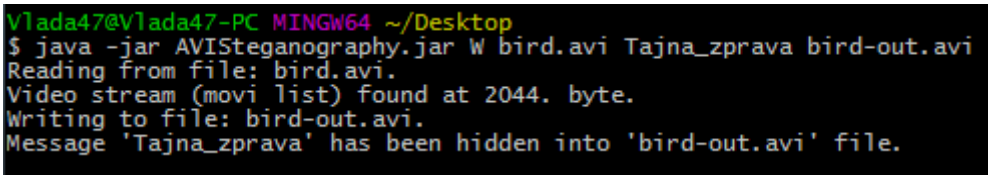
Uživatelská dokumentace

Program lze použít v jednom ze dvou následujících způsobů:

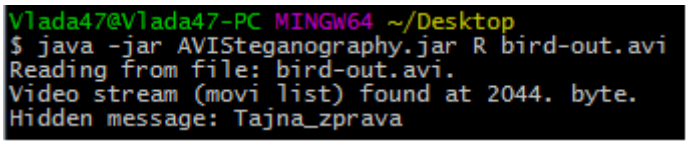
- **Ukrytí zprávy do souboru**
  - `java -jar AVISteganography W <vstupní_soubor> <zpráva> <výstupní_soubor>`
  - příklad – obr. 1
- **Nalezení zprávy v souboru**
  - `java -jar AVISteganography R <vstupní_soubor>`
  - příklad – obr. 2

Pokud uživatel zadá špatně nějaké parametry, případně jich zadá nedostatečný počet, dostane varování a nápovědu ke spuštění programu (obr. 3).

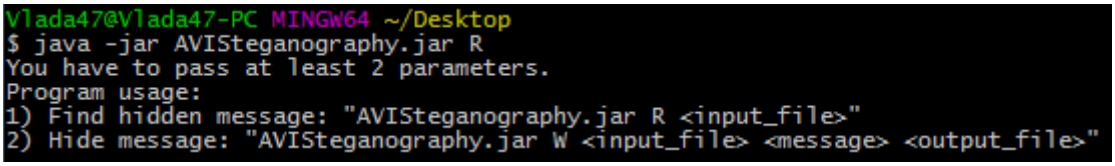
V případě, že soubor není formátu AVI, uživatel je informován o nenalezení hledané sekvence s datovým proudem (video/audio).



Obrázek 1: uložení zprávy do souboru



Obrázek 2: nalezení zprávy v souboru



Obrázek 3: nedostatečný počet parametrů

Závěr

Výsledné řešení je poměrně dobrým konceptem pro steganografii textu v souboru formátu AVI. Umožňuje ukrýt teoreticky i velmi dlouhé zprávy bez viditelného vlivu na vzhled obrazu obsaženého videa nebo poškození důležitých částí souboru. Zároveň je možné provést poměrně jednoduché rozšíření, které by umožnilo skrýt zprávu na specifické místo v rámci datového proudu (lze při hledání zprávy identifikovat pomocí úvodní sekvence) nebo ukrýt jiný soubor jakéhokoliv obsahu (načteného do pole bytů), pokud by nepřesahoval jistou velikost.

Obecně vzato by bylo možné program použít jako základ pro určitou platformu steganografie do souborů s multimediálním obsahem.

Bibliography

AVI RIFF File Reference [https://msdn.microsoft.com/en-us/library/windows/desktop/dd318189(v=vs.85).aspx]