INF FAKULTÄT FÜR INFORMATIK

StegoDetect:
Steganography und verdeckte Kommunikation - Attributierung

SMKITS
09.11.2022

Bernhard Birnbaum
Ulrich Hasler

Inhalt

1. Motivation
2. Aufgabenverständnis
    a. Inhaltlich
    b. SMK-Aspekt
3. Fortschritte
    a. Aufsetzen der Arbeitsumgebung
    b. Einarbeiten in Materialien
    c. Zusammenstellen des Bildtestsets
4. Probleme bei der Umsetzung
5. Quellen

1. Motivation

- steganographische Methoden finden Anwendung in moderner Schadsoftware
- zur Prävention ist neben der Detektion auch Attributierung der Angreifer wichtig
- es werden Sicherheitsaspekte verletzt (Vertraulichkeit und Authentizität)

2. Aufgabenverständnis: Inhaltlich

- durch Analyse von bekannten Einbettungsmethoden auf Möglichkeiten und Grenzen individueller Einbettungsparameter schließen
    - Einbettungsschlüssel
    - Einbettungskapazitäten
    - Einbettungsinhalte
- Untersuchung und Evaluation der Parameterwahl, ob und wie diese Möglichkeiten der Attributierung bieten

2. Aufgabenverständnis: SMK-Aspekt

- pro Woche ein Meeting mit TaskCoach und ein internes Gruppenmeeting
- Protokoll jede Woche für
    - Nachbesprechung der Aufgaben
    - Probleme und Fragen
    - Aufgaben bis zum nächsten Meeting
    - Minutes of Meeting

3. Fortschritte: Aufsetzen der Arbeitsumgebung

- Installation von Docker
- Pull des Dockercontainers von GitHub “DominicBreuker/stego-toolkit”
- Zusammenstellen eines Bildtestsets (angefangen)

3. Fortschritte: Einarbeiten in Materialien

- Umgang mit Tools
    - Steganography-Tools (jphide, jsteg, outguess, …)
    - Stegoanalysis-Tools (stegdetect, stegoVeritas, …)
    - Werkzeuge sind im Stego-Toolkit zusammengefasst
- Einlesen in Paper zur Attributierung anhand von statistischen Bildmerkmalen

3. Fortschritte: Zusammenstellen des Bildtestsets

- Bildtestset muss verschiedene Attributierungsmerkmale abdecken
    - statistische Merkmale (Auflösung, Größe, …)
    - inhaltsbasierte Merkmale (Differenzbild, Kanten, …)
- Diversität der Bilder entscheidend
    - schwarz/weiß bzw. Farbbilder
    - unterschiedliche Auflösungen (z.B. Bilder von Handy-Kamera)
    - unterschiedliche Motive (z.B. Landschaften, Nahaufnahmen, …)
- Bildtestset besteht ausschließlich aus JPEG-Bildern
    - meistgenutztes Format für Einbettungen aufgrund der Art der Kompression (DCT)
    - Stego-Toolkit auf JPEG-Bilder ausgerichtet

4. Probleme bei der Umsetzung

- tote Links für Cover-Datenbanken in Referenzen (BOSS)
- Bilder von BOWS2 im PGM-Raw-Format → Tools unterstützen nur JPEG bzw. PNG-Bilder
  - → Umwandeln der PGM-Bilder in JPEG-Format
  - → zusätzliche alternative Datenbank: Kaggle/Alaska2
- Docker-Einrichtung auf PopOs zunächst fehlerhaft → Neuaufsetzen nötig

5. Quellen

- Stego-Toolkit
    - https://github.com/DominicBreuker/stego-toolkit
- Detecting Steganographic Content on the Internet
    - http://www.citi.umich.edu/u/provos/papers/detecting.pdf
- Break Out Watermarking System 2 (BOWS2)
    - http://bows2.ec-lille.fr
- Kaggle/Alaska2 Datenbank
    - https://www.kaggle.com/competitions/alaska2-image-steganalysis/data