INF FAKULTÄT FÜR INFORMATIK

StegoDetect:
Steganographie und verdeckte Kommunikation - Attributierung

SMKITS
14.12.2022

Bernhard Birnbaum

Inhalt

1. Fortschritte/Projektentwicklung
2. Konzept
    a. finale Werkzeugauswahl
    b. Attributierungsmerkmale
    c. Testprotokoll
3. Umsetzung
    a. Bildtestset
    b. Shell-Script
4. Probleme
5. Aussicht
    a. Draft
    b. nächste Schritte
6. Quellen & Fragen

1. Fortschritte: Aufgabenstellung

(5) StegoDetect: Steganography und Verdeckte Kommunikation: Attributierung - CK und JD – 3-6 Studierende

Aufgabenverteilung im Team:

- Erstellung eines Original-Bildtestsets (Coverdaten) - **alle**

DR1

- Recherche nach Bildmerkmalen zur Unterscheidung (Attributierung): statistische Bildmerkmale, inhaltsbasierte Merkmale (Differenzbild, Kanten usw.) usw. und tabellarische Zusammenfassung sowie Auswahl an Werkzeugen/Programmcode zur Analyse - **alle**
- **Individueller Teil**: Auswahl pro Teammitglied von mind. zwei Bildstego-Verfahren in allen 3 Variationen – ggf. Rücksprache mit Task Coach/JD zur individuellen Aufteilung halten, was wie sinnvoll wird
- Erarbeitung eines Testprotokolls (Tabelle und Ablaufdiagramm) für die Testziele:
  - (1) Variation von Schlüssel/Password unter Beachtung von kurzen und langen Schlüssel und des kompletten Schlüsselraums und
  - (2) Variation des Einbettungstextes/Payload (kurz, lang) sowie deren
  - (3) Kombinationen Schlüssel/Password-Payload sowie einschließlich Qualitätssicherungsmaßnehmen (Einbettung- und Auslesen erfolgreich plus Steganalysis erfolgreich oder nicht) im Intra- und Inter-Stegoverfahrenvergleich und Intra- und Intermedienvergleich – **alle** (ggf. Reduktion auf ein sinnvolles Maß – begründete Auswahl)
- Auswahl an zu nutzenden Stego-Verfahren für jeden Teammitglied – Ausgangspunkt: jphide/jpseek, jsteg, openstego, outguess, Steghide, F5, sowie Auswahl an Detektionsansätzen zur Qualitätssicherung der Einbettung pro Teammitglied: stegoVeritas, zsteg und stegdetect – **alle, dann individuell jedes Teammitglied**
- Erstellung von Cover-Stego-Datenpaaren mit den zu testenden Variationen aus dem Testprotokoll und dazugehörigen Metadaten (Annotationen/Testinformationen) und Infos zur Qualität: Auslesen erfolgreich/Detektion erfolgreich/nicht erfolgreich pro Teammitglied für die gewählten Verfahren - **individuell**
- Auswahl, Umsetzung und Analyse von Bildmerkmalen zur Unterscheidung (Attributierung) auf Basis der tabellarischen Zusammenfassung (alle) für die Cover-Stego-Paare in den Variationen (1)-(3) pro Verfahren (Intra-Verfahrensattributierung) und Inter-Verfahrensattributierung in Intra-Bild und Inter-Bild Vergleichen - **individuell**

DR2

- Detailanalyse der Stego-Cover-Daten vor den Testzielen (Variationen) vor den ausgewählten zu untersuchenden Bildmerkmalen - **individuell**
- Umsetzung und Untersuchung sowie Dokumentation und Bewertung der betrachteten Testfälle - **individuell**
- Darstellen der Ergebnisse im Intra- und Inter-Verfahren- und Intra-/Intermedien-Vergleich – **individuell und alle**

- Fortschritte seit DR1
- Konzept und Umsetzung weitestgehend abgeschlossen
- Analyse, Auswertung und Report ausstehend

1. Fortschritte: Projektentwicklung

- einseitige Kommunikation und fehlende Initiative
- seit DR1 alleine weitergearbeitet
    - vor zwei Wochen: letzte Möglichkeit der Einbringung in Projekt
    - vor einer Woche: endgültiger Ausstieg von Teampartner aus Projekt
- mögliche Anpassung der Aufgabenstellung in Absprache mit TaskCoach:
    - Werkzeuge jphide und stegbreak werden vernachlässigt (dazu später)
    - Untersuchung des kompletten Schlüsselraums auf kurzen und langen Schlüssel beschränken (Codeanalyse der Tools wäre nötig → zu zeitaufwändig)
    - (maximal mögliche) relative Einbettungsdatenlänge wird nicht untersucht
    - Beschränkung der Attributierungsmerkmale aufgrund von weniger detaillierter Analyse, wie sie mit zwei Leuten möglich gewesen wäre (inhaltsbasierte Merkmale werden vernachlässigt)

    → Verweise auf weiterführende Untersuchungen im Report

2. Konzept: finale Werkzeugauswahl

- Stego-Tools
  - jsteg
  - outguess, outguess-0.13
  - steghide
  - f5 (Ausführung sehr langsam → Bilder mit Dimensionsgröße > 1024 werden übersprungen)
- Analyse-Tools
  - exiftool, binwalk, strings, foremost
  - imagemagick (identify für Metadaten, compare für Differenzbilderstellung)
  - stegoveritas (Ausführung sehr langsam → Bilder mit Dimensionsgröße > 1024 werden übersprungen)
  - stegdetect
  - stegbreak (Ausführung in ~10% erfolgreich)

2. Konzept: Attributierungsmerkmale

- generell: Einbettung und Extraktion erfolgreich?
- statistische Merkmale
    - Bildformat/MINE-Type
    - JFIF Version und Encoding
    - Dateigröße
    - verwendete Kamera
    - Auflösung
    - Header
    - Foremost-Datenextraktion, StegDetect/StegBreak Ergebnis
- inhaltsbasierte Merkmale
    - Differenzbild mit Original: Verhältnis SW (Stego-Bild, stegoveritas-Bilder: Betrachtung einzelner Farbkanäle und Kanten)

2. Konzept: Testprotokoll (Tabelle)

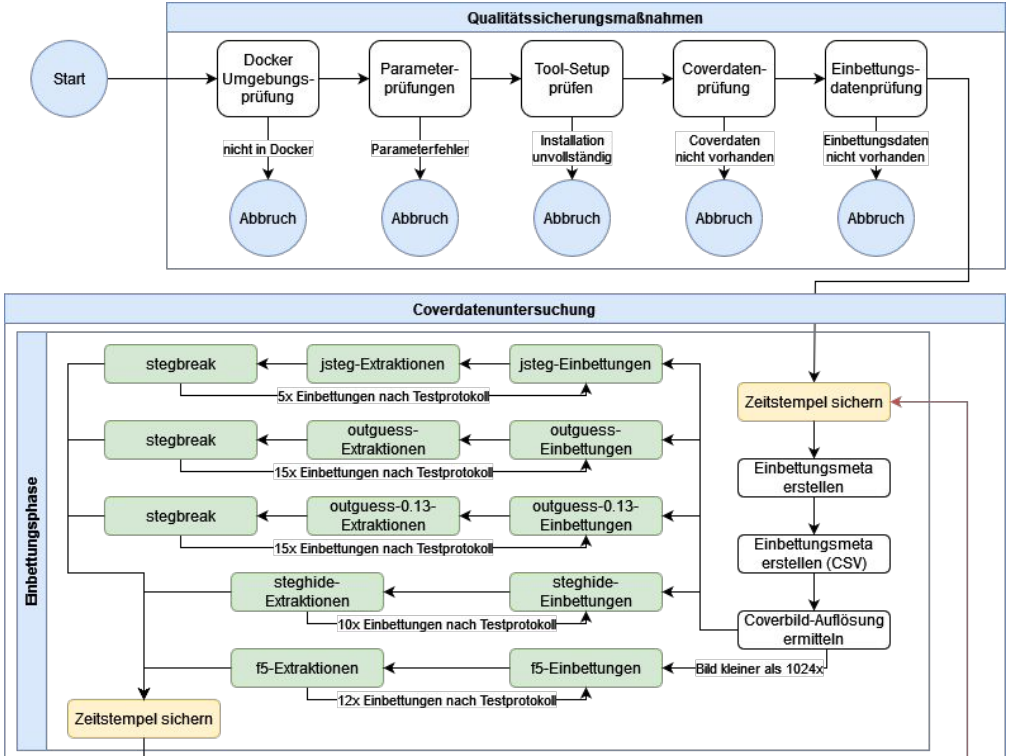
Einbettungsschlüssel	Einbettungsdaten	nicht-unterstützte Tools
kein Schlüssel	kurze Einbettung	jphide, steghide
kein Schlüssel	mittellange Einbettung	jphide, steghide
kein Schlüssel	lange Einbettung	jphide, steghide
kein Schlüssel	Einbettung mit geringer Entropie	jphide, steghide
kein Schlüssel	binäre Einbettung	jphide, steghide, f5
kurzer Schlüssel	kurze Einbettung	jsteg
kurzer Schlüssel	mittellange Einbettung	jsteg
kurzer Schlüssel	lange Einbettung	jsteg
kurzer Schlüssel	Einbettung mit geringer Entropie	jsteg
kurzer Schlüssel	binäre Einbettung	jsteg, f5
langer Schlüssel	kurze Einbettung	jsteg
langer Schlüssel	mittellange Einbettung	jsteg
langer Schlüssel	lange Einbettung	jsteg
langer Schlüssel	Einbettung mit geringer Entropie	jsteg
langer Schlüssel	binäre Einbettung	jsteg, f5

- jsteg: 1x5 → 5 Einbettungen
- outguess: 3x5 → 15 E.
- outguess-0.13: 3x5 → 15 E.
- steghide: 2x5 → 10 E.
- f5: 3x4 → 12 E.

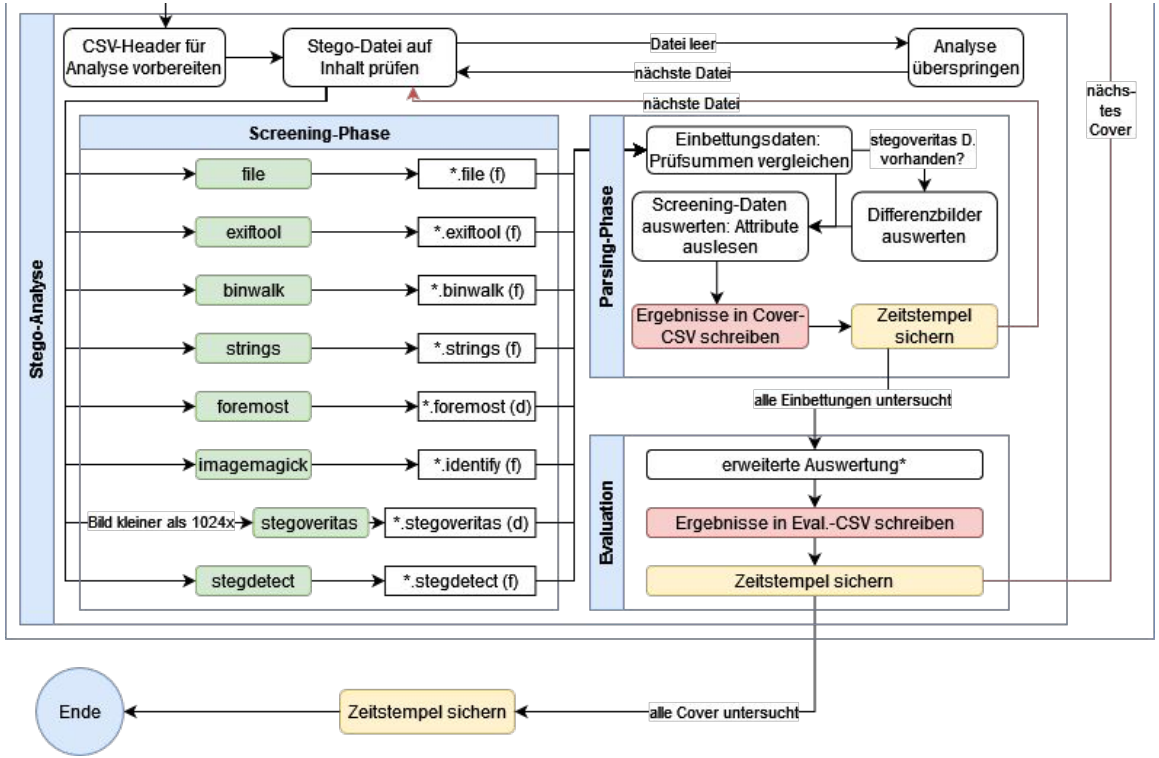
→ 57 Einbettungen, wenn Bild > 1024x → 45 Einbettungen

- Schlüssel: 4 bzw. 50 Byte
- Einbettungsdaten: 67 Byte (kurz) bis 17.5 KB (lang)

2. Konzept: Testprotokoll (Diagramm) I



2. Konzept: Testprotokoll (Diagramm) II



3. Umsetzung: Bildtestset

- 640x Bilder aus Kaggle/Alaska2 Datenbank (Farbbilder, 512x512), zufällige Auswahl
- 192x Bilder aus BOWS2-Datenbank (Schwarz-Weiß-Bilder, 512x512), zufällige Auswahl
- 192x Bilder aus privater Quelle mit verschiedenen Kameras und Auflösungen

→ 1024 Bilder insgesamt

3. Umsetzung: Shell-Script

- Funktionalitäten (durch Parameter steuerbar)
    - Erstellung der Cover-Stego-Paare mit Stego-Tools
    - Auswerten der Cover-Stego-Paare mit Analyse-Tools
    - Parsen der Programmausgaben → CSV-Tabelle
    - Evaluation des Covers pro Stego-Tool (Durchschnittswerte, erfolgreiche/fehlerhafte Einbettungen, …)
    - zusätzliche Untersuchungsfunktion für einzelne Bilder auf Basis der Untersuchungsergebnisse für Attributierung
- Ausführungsdauer
    - je nach Bild zwischen 3 und 12min im Worst-Case, ~5-8min/Bild

4. Probleme I

- Script: Speicherplatzproblem durch Analysedaten
    - pro Bild entstehen ~250 MB an Daten → für 1024 Bilder → 250 GB
    - → Löschen der Analysedaten nach Auswertung jedes Covers, nur CSV mit Ergebnissen wird behalten
- Script: Ausführungszeit
    - durchschn. 5-8min pro Bild → für komplettes Cover-Set ~4d
    - Möglichkeit Cluster-Ausführung mit SLURM Jobscript
    - → Probleme: Docker-Installation, Registrierung, keine ARM-Architektur!
    - → wahrscheinlich ist der Aufwand für Parallelisierung höher als zeitliches Ersparnis
- Script: inhaltsbasierte Analyse
    - automatisierte Inhaltsbetrachtung der Bilder möglich
    - → Untersuchungen weitestgehend unabhängig von Bildinhalt

4. Probleme II

- jphide/jpseek
    - Problem: Schlüsseleingabe ist interaktiv → keine Automatisierung möglich

    → neuen Parameter in Quelltext eingefügt und statische Neukompilierung

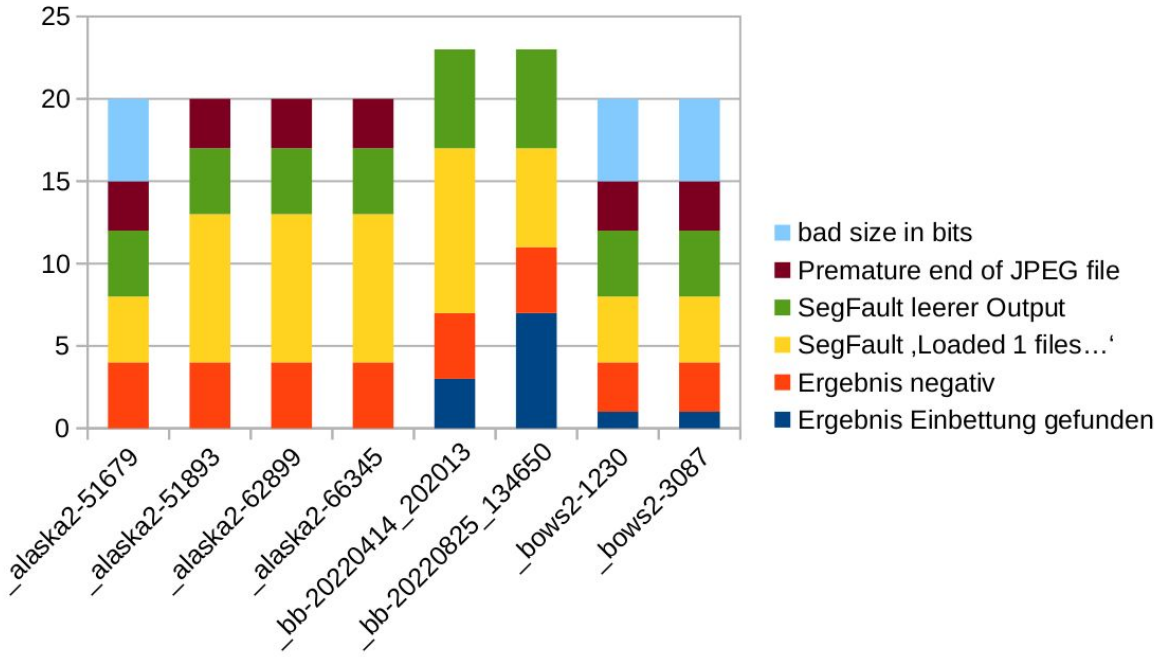
    → immer SegFault-Error (Fehlercode 139) → Fehler tritt bereits vor eigener Manipulation auf

    → Tool in Absprache mit TaskCoach abgeschrieben
- stegbreak
    - Problem: immer SegFault-Error (Fehlercode 139)

    → Neukompilierung gefixter Version in Docker-Umgebung

    → SegFault-Fehler in ~75% der Fälle

4. Probleme III



- alaska2 und BOWS-Bilder: immer die gleichen Kombinationen von Tool/Einbettungs-parametern schläft fehl bzw. laufen durch
- private Bilder: mehr Ausführungen, da outguess öfter erfolgreich einbetten konnte

5. Aussicht: Draft

5. Aussicht: nächste Schritte

- Detailanalyse: evtl. identifizieren weiterer sinnvoller Attribute
- Dokumentation und Bewertung der Testfälle
- Darstellen der Ergebnisse im Intra- und Inter-Verfahren- bzw. Intra- und Inter-Medien-Vergleich

→ Verbesserung der Tool- und Einbettungs-Attributierung

- Report schreiben

6. Quellen

- Detecting Steganographic Content on the Internet
  - http://www.citi.umich.edu/u/provos/papers/detecting.pdf
- Forensic data model for artificial intelligence based media forensics
  - https://doi.org/10.2352/EI.2022.34.4.MWSF-324
- Attributierung
  - http://www.guillermito2.net/stegano/jsteg/index.html
- Tools
  - https://github.com/DominicBreuker/stego-toolkit
  - https://github.com/abeluck/stegdetect
  - https://github.com/h3xx/jphs
  - https://imagemagick.org/
- Bildtestset
  - http://bows2.ec-lille.fr
  - https://www.kaggle.com/competitions/alaska2-image-steganalysis/data