NIST
National Institute of Standards and Technology
U.S. Department of Commerce

FORENSICS @ NIST

#NISTForensics

Approximate Matching: Testing how well matchers work

Team
Monika Singh
Douglas White
Barbara Guttman

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Disclaimer

Trade names and company products are mentioned in the text or identified. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

Motivation

Motivation

- In today's world everything is going digital.
  - Conventional books have been converted to eBooks.
  - Letters have been converted to emails.
  - Images have been digitized (jpeg, png).
  - Compact cassettes and video cassettes have been converted to mp3 and mp4s.

Motivation

- In today's world everything is going digital.
  - Conventional books have been converted to eBooks.
  - Letters have been converted to emails.
  - Images have been digitized (jpeg, png).
  - Compact cassettes and video cassettes have been converted to mp3 and mp4s.
- It is an enormous volume of data for a forensic investigator to manually examine in a reasonable period of time.

Motivation

1 terabyte 1 terabyte of digital text is (approximately) equal to:

- 1 trillion characters: 1 character = 1 byte.
- 220 million pages: 1 page = 5000 characters.
- 21 years of printing time: 20 sheets per minute.
- 1 million kg of paper: onesided printed.
- Paper stack of 22 km height: bulk of 0.1 mm.

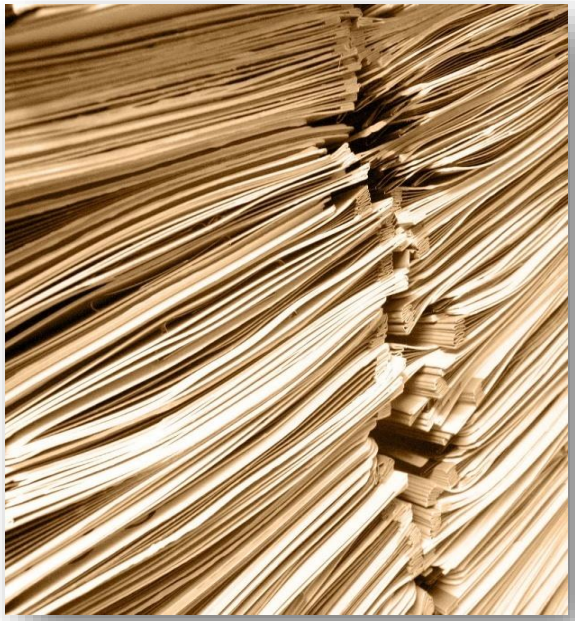
Motivation

1 terabyte 1 terabyte of digital text is (approximately) equal to:

- 1 trillion characters: 1 character = 1 byte.
- 220 million pages: 1 page = 5000 characters.
- 21 years of printing time: 20 sheets per minute.
- 1 million kg of paper: onesided printed.
- Paper stack of 22 km height: bulk of 0.1 mm

Motivation

A major requirement of modern digital forensic investigations is an ***automatic filtering*** of the correlated/relevant data, that otherwise requires a manual examination by the investigator.



Automatic Filtering

Motivation

Filtering can be of following types:

- Whitelisting
  - The process filters out files by matching them with the database of already ***known to be good*** files.
  - Matched files do not require manual investigation by the investigator.
- Blacklisting
  - The process filters out files by matching them with the set of already ***known to be bad*** files.
  - Matched files are malicious files that the investigator needs to examine further.

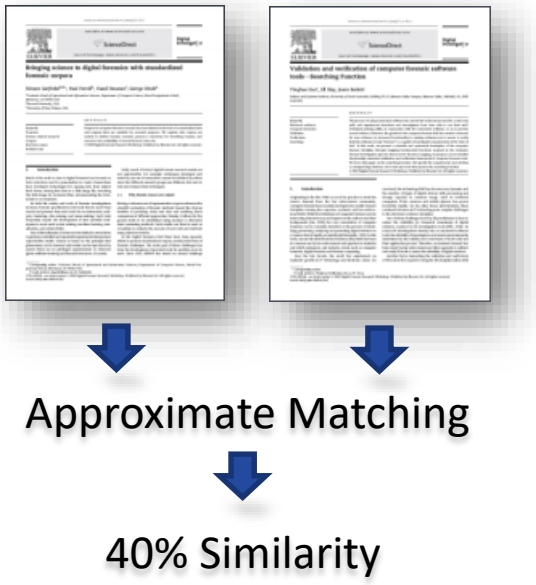
Motivation

The Cryptographic hash functions in digital forensics

- Identify exact duplicates.
- However, they fail to detect similarity.
  - If the input of the hash function is changed slightly (say, the flip of a single bit), the output changes significantly (half of the output bits would get flipped).
- The investigators need robust algorithms that allow similarity detection.

Approximate Matching

The approximate matching is a generic term describing any technique designed to identify similarity between two digital artifacts.



Approximate Matching

- Standard definition, terminology and essential requirement of an Approximate Matching algorithm has been defined by NIST (January 2014).
  - DRAFT NIST SPECIAL PUBLICATION 800-168 "APPROXIMATE MATCHING: DEFINITION AND TERMINOLOGY"
- Not much security analysis of the existing algorithm has been done.

Approximate Matching

Existing Schemes:

- dcfldd (Nicholas Harbour, 2002)
- ssdeep (Jesse Kornblum, 2006)
- sdhash (Vassil Roussev, 2010)
- bbHash(Frank Breitinger 2012)
- mvHash-B(Frank Breitinger 2013)
- mrsh-v2(Frank Breitinger 2013)

Evaluation Framework

To develop a suitable evaluation framework, based on which the evaluation of existing approximate matching algorithms and upcoming schemes can be performed.

Use Cases:

An approximate matching algorithm should address at least one of the following problems:

- **Embedded Object Identification**
- **Fragment Identification**
- **Related Document Detection**
- **Code Version Identification**

Use Cases:

An approximate matching algorithm should address at least one of the following problems:

- **Embedded Object Identification**
  - Identify a given object inside an artifact.
  - Identify artifacts that share a common object
- **Fragment Identification**
- **Related Document Detection**
- **Code Version Identification**

Use Cases:

An approximate matching algorithm should address at least one of the following problems:

- **Embedded Object Identification**
  - Identify a given object inside an artifact.
  - Identify artifacts that share a common object
- **Fragment Identification**
  - Identify the presence of traces/fragments of a known artifact, e.g., identify the presence of a file in a network stream based on individual packets.
- **Related Document Detection**
- **Code Version Identification**

Use Cases:

An approximate matching algorithm should address at least one of the following problems:

- **Embedded Object Identification**
  - Identify a given object inside an artifact.
  - Identify artifacts that share a common object
- **Fragment Identification**
  - Identify the presence of traces/fragments of a known artifact, e.g., identify the presence of a file in a network stream based on individual packets.
- **Related Document Detection**
  - Identify related artifacts, e.g., different versions of a document.
- **Code Version Identification**

Use Cases:

An approximate matching algorithm should address at least one of the following problems:

- **Embedded Object Identification**
  - Identify a given object inside an artifact.
  - Identify artifacts that share a common object
- **Fragment Identification**
  - Identify the presence of traces/fragments of a known artifact, e.g., identify the presence of a file in a network stream based on individual packets.
- **Related Document Detection**
  - Identify related artifacts, e.g., different versions of a document.
- **Code Version Identification**
  - Software version, malware detection

Use Cases:

- **Existing Schemes:**
  - ssdeep (Jesse Kornblum, 2006)
  - sdhash (Vassil Roussev, 2010)
  - bbHash(Frank Breitinger 2012)
  - mvHash-B(Frank Breitinger 2013)
  - mrsh-v2(Frank Breitinger 2013)

Use Cases:

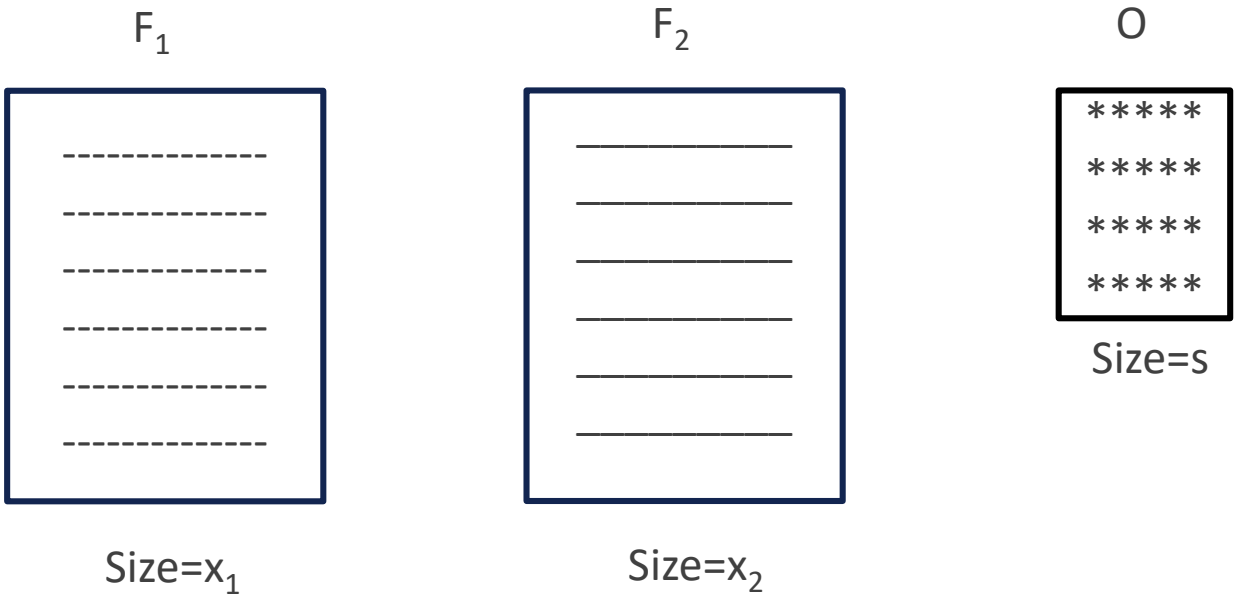
- **Existing Schemes:**
  - ssdeep (Jesse Kornblum, 2006)
  - sdhash (Vassil Roussev, 2010)
  - bbHash(Frank Breitinger 2012)
  - mvHash-B(Frank Breitinger 2013)
  - mrsh-v2(Frank Breitinger 2013)
    - Encase
    - FTK
    - X-Ways

Embedded Object Identification

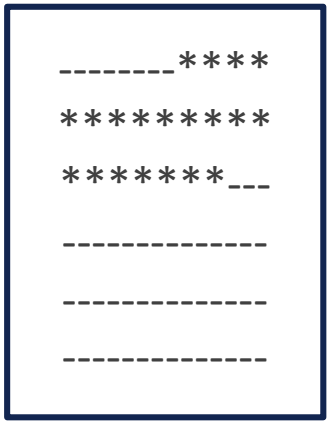
Dataset Generation:

- Dataset type: DOCX, PPTX
- Embedded Object: jpeg, bmp, gif, tiff
- Procedure:
  - Embed each object one by one in each target file at randomly chosen position.
  - For example:
  - User provide 10 docx files(target file) and 10 jpeg (object)
  - Generated embedded file: 100

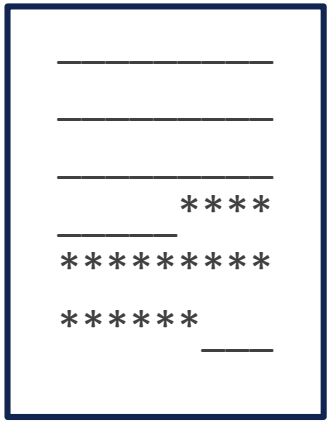
Embedded Object Identification



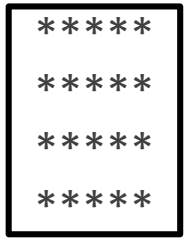
Embedded Object Identification



Size=x_1+s



Size=x_2+s



Size=s

Embedded Object Identification

Sub Test cases:

- **Embedded Object Detection**

  Identify a given object inside a file.

- **Single Common Object Identification**

  Identify files that share a common object (cross object identification)

Embedded Object Identification

Measures:

- True Positive
- True Negative
- False Positive
- False Negative
- False positive rate (FPR)
- False negative rate (FNR)
- **Precision** : A perfect precision score of 1.0 means that every result retrieved by a search was relevant.
- **Recall** : A perfect recall score of 1.0 means that all relevant documents were retrieved by the search.
- **F-score** : A measure of a test's accuracy (best value at 1 and worst at 0)
- **MCC** : A measure of the quality of Test(best value at 1 and worst at -1)

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Embedded Object Identification

Fragment Identification

Dataset Generation:

- Dataset type: Text, Docx
- Fragment size:
  95%,90%,85%,80%,……5%,4%,3%,2%,1%.

Fragment Identification

Dataset Generation:

- It sequentially cuts X% of the original input length and generates the match score where X = 5 by default.
- For example file size= 100,000 bytes

A s d f g h j . . . k l p o I u y t r e . . . v b n m x z . . . . a s l k j h . . . s a q w e r t y u l o g . . . h j j k l p

Fragment Identification

Dataset Generation:

- Maximum cuts: $\lceil \frac{100}{x} \rceil$-1
- So for a 100000 bytes long document there will be total 19 cuts of 5000 bytes.

Fragment Identification

Dataset Generation:

- In case that the algorithm still identifies similarity we continue with a further reduction in 1% steps until only 1% of the input is left.
- So the algorithm continues cutting in 1000 byte long pieces until only 1% of the input is left.

Fragment Identification

Dataset Generation:

- **Random fragment** is the first mode. The framework randomly decides whether to start cutting at the beginning or the end of an input and then continues randomly.
- **Sequential Fragment** is the second mode and only cuts blocks at the beginning of an input.

Fragment Identification

Test cases:

- **Fragment detection** identifies similarity tools ability to correlate a an input and a fragment.
- **Smallest Fragment Correlation test** identifies what is the smallest piece or fragment, for which the similarity tool reliably correlates the fragment to the original file?

Fragment Identification

- **Measures:**
  - True Positive
  - True Negative
  - False Positive
  - False Negative
  - False positive rate (FPR)
  - False negative rate (FNR)
  - **Precision** : A perfect precision score of 1.0 means that every result retrieved by a search was relevant.
  - **Recall** : A perfect recall score of 1.0 means that all relevant documents were retrieved by the search.
  - **F-score** : A measure of a test's accuracy (best value at 1 and worst at 0)
  - **MCC** : A measure of the quality of Test(best value at 1 and worst at -1)

Fragment Identification

Thank You ☺

We are also doing a demo during the poster session. Please stop by if you want to know more about our tool.