Data Integrity Fingerprint (DIF)

A proposal for a human-readable fingerprint of scientific datasets that allows verifying their integrity

O. Lindemann & F. Krause

Date: 12 December 2021

Oliver Lindemann (oliver@expyriment.org) & Florian Krause (florian@expyriment.org)

Introduction

Problem:
How can we link a journal article unmistakably and indefinitely to a related (open) dataset, without relying on storage providers or other services that need to be maintained?

Solution:
The author calculates checksums of all the files in the dataset the article relates to. From these checksums the author calculates the *Data Integrity Fingerprint (DIF)* - a single "master checksum" that uniquly identifies the entire dataset. The author reports the DIF in the journal article. A reader of the journal article who obtained a copy of the dataset (from either the author or any other source) calculates the DIF of their copy of the dataset and compares it to the correct DIF as stated in the article. If the list of checksums of individual files in the original dataset is available, the author can furthermore investigate in detail the differences between the datasets, in case of a DIF mismatch.

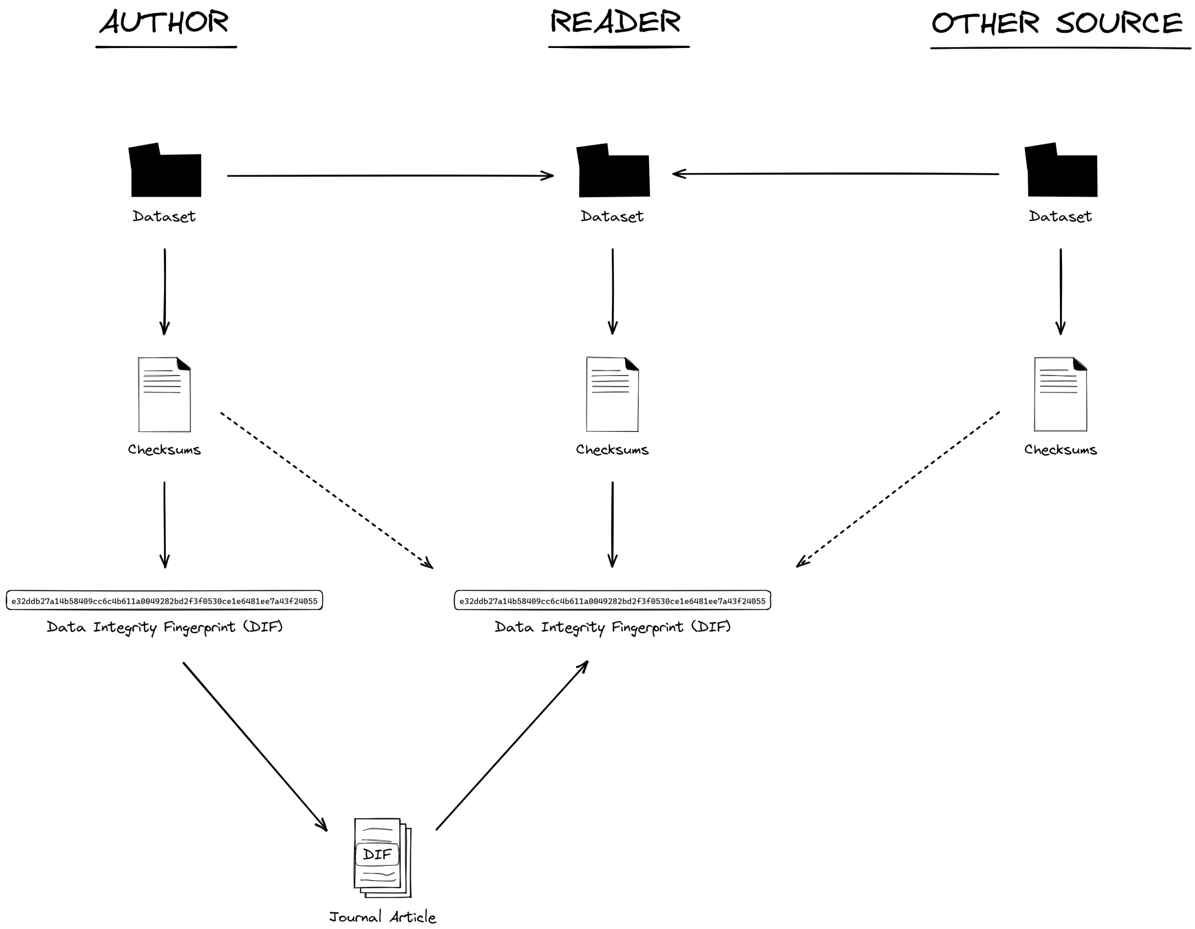
AUTHOR
READER
OTHER SOURCE
Dataset
Dataset
Dataset
Checksums
Checksums
Checksums
e32ddb27a14b58409cc6c4b611a0049282bd2f3f0530ce1e6481ee7a43f24055
Data Integrity Fingerprint (DIF)
e32ddb27a14b58409cc6c4b611a0049282bd2f3f0530ce1e6481ee7a43f24055
Data Integrity Fingerprint (DIF)
DIF
Journal Article

Figure 1: DIF_Procedure_Flowchart

Procedure for calculating the DIF of a dataset

1. Choose a (cryptographic) hash function `Hash` (e.g. SHA-256)
2. For every file `f` in the (potentially nested) subtree under the dataset root directory (with symbolic links being followed),
    - calculate the checksum `c` as the hexadecimal digest (lower case letters) of `Hash(f)` (i.e. the hashed *binary contents* of the file)
    - get the file path `p` as the UTF-8 encoded relative path in Unix notation (i.e. U+002F slash character as separator) from the dataset root directory to `f`
    - create the string `cp` (i.e the concatenation of `c` and `p`)
    - add `cp` to a list `l`
3. Sort `l` in ascending Unicode code point order (i.e., byte- wise sorting, NOT based on the Unicode collation algorithm)
4. Create the string `l[0]l[1]...l[n]` (i.e. the concatenation of all elements of `l`)
5. Retrieve the DIF as the hexadecimal digest of `Hash(l[0]l[1]...l[n])`

Optionally, checksums of individual files and their file paths can be saved as a checksums file with lines of `c␣␣p` for each `f` (i.e. `c` followed by two U+0020 whitespace characters followed by `p`).

Available implementations

- Python (reference implementation): dataintegrityfingerprint-python
- further implementations coming soon

Note: On a GNU/Linux system with a UTF-8 locale, the procedure to create the SHA-256 DIF is equivalent to:

```
cd <DATASET_ROOT_DIRECTORY>
export LC_ALL=C
find -L . -type f -print0 | xargs -0 shasum -a 256 | sed 's/^\\*//;s/\\\\*/\\/' |\
            cut -c-64,69- | sort | tr -d '\n' | shasum -a 256 | cut -c-64
```

Example data

Custom implementations may be tested against example data to verify correctness.

Discussion

For comments and remarks about this proposal, please use the Discussions forum of our Github repository.