# 1 Introduction

One of the biggest challenges facing digital forensic investigation is coping with the vast number of files to be processed. Hashing algorithms have become an indispensable part of computer science since their output - hashes - can be used as unique identifiers to compare many digital artifacts. The best known are the so-called cryptographic hashes, with the help of which only absolutely identical original images can be mapped to the same hash. This makes it possible to efficiently verify the integrity of these original images. If a bit in one of the original images changes, they map to different hashes. Cryptographic hashes are therefore unsuitable for recognizing merely similar artifacts. The so-called cryptographic diffusion obscures the relationship between ciphertext and plaintext as it is intended for one-way functions. Fuzzy hashes break the cryptographic diffusion and still hide the relationship between the original image and the hash. Two fuzzy hashes are similar to each other to the same extent as their originals are similar. This results in a variety of practical application cases where fuzzy hashes can be used. According to Roussev [24], the most promising applications of fuzzy hashes can be abstracted as follows:

- Document similarity detection: identify related documents, e.g., different versions of a Word document.
- Embedded object detection: identify a given object inside a container, e.g., a JPG within a Word document.
- Fragment detection: identify an original input based on a fragment, e.g., analyzing a device on the byte level.
- Clustering files: group files that share similar content, e.g., a Word document and an e-mail.

As of late, fuzzy hashes have become indispensable in static malware detection and are one of the cornerstones in Google's Virustotal [7]. Through so-called approximate matching,