HOCHSCHULE MITTWEIDA
University of Applied Sciences

BACHELORARBEIT

Frau
Raika Käbisch

Analyse digitaler Literatur hinsichtlich Identifikationsmerkmalen

Mittweida, September 2023

Fakultät **Angewandte Computer- und Biowissenschaften**

BACHELORARBEIT

Analyse digitaler Literatur hinsichtlich Identifikationsmerkmalen

Autorin:
Raika Käbisch

Studiengang:
Allgemeine und Digitale Forensik

Seminargruppe:
FO20w5-B

Erstprüfer:
Prof. Dr. rer. nat. Dirk Labudde

Zweitprüfer:
M.Sc. Felix Fischer

Einreichung:
Mittweida, 01.09.2023

Verteidigung/Bewertung:
Mittweida, 2023

Faculty of **Applied Computer Sciences and Biosciences**

BACHELOR THESIS

Analysis of digital literature with regard to identification features

Author:
Raika Käbisch

Course of Study:
General and Digital Forensics

Seminar Group:
FO20w5-B

First Examiner:
Prof. Dr. rer. nat. Dirk Labudde

Second Examiner:
M.Sc. Felix Fischer

Submission:
Mittweida, 01.09.2023

Defense/Evaluation:
Mittweida, 2023

Bibliografische Beschreibung:

Käbisch, Raika:
Analyse digitaler Literatur hinsichtlich Identifikationsmerkmalen. – 2023. – 50 S.
Mittweida, Hochschule Mittweida – University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2023.

Referat

In der vorliegenden Arbeit werden verschiedene Arten von Dokumenten dahingehend überprüft, ob sich darin digitale Wasserzeichen befinden. Dabei liegt der Schwerpunkt auf dem Unterschied, wenn diese Dokumente von zwei verschiedenen Benutzerkonten heruntergeladen werden. Es soll geklärt werden, ob Unterschiede aufkommen und wo sich diese befinden. Zuerst werden theoretische Grundlagen verdeutlicht, um das Ergebnis einordnen zu können. Anschließend wird sich auf verschiedene Arten von Dokumenten festgelegt, welche untersucht werden sollen. Für die Untersuchung werden die Dokumente einzeln über verschiedene Benutzerkonten heruntergeladen, anschließend die Hashwerte berechnet und verglichen. Werden Unterschiede innerhalb der Hashwerte deutlich, soll eine weitere Untersuchung auf Byteebene durchgeführt werden, um herauszufinden, wo sich die Dateien unterscheiden. Hierdurch ist es möglich festzustellen, dass einige Dokumentarten Wasserzeichen enthalten, welche durch die Benutzerkonten generiert werden. Die vorliegende Arbeit zeigt, dass digitale Wasserzeichen bereits in einer Vielzahl an Dokumenten verwendet werden, gleichzeitig jedoch noch viel Potenzial tragen, um zukünftig erstellte Dokumente noch besser zu schützen und den Urheber sowie das Benutzerkonto nachweisen zu können.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

DOI . . . . . Digital Object Identifier

DS-SS . . . . . Direct Sequence Spread Spectrum

DWT . . . . . diskrete Wavelet-Transformation

FH-SS . . . . . Frequency Hopping Spread Spectrum

ISBN . . . . . International Standard Book Number

LSB . . . . . Least Significant Bit

PDF . . . . . Portable Document Format

PRNS . . . . . Pseudo Random Noise Sequence

rtf . . . . . Rich Text Format

SSW . . . . . Spread Spectrum Watermarking

URN . . . . . Uniform Resource Name

1 Einleitung

Laut einer Studie der Polizeilichen Kriminalstatistik (PKS), ist die Anzahl der erfassten Straftaten im Zusammenhang mit Urheberrechtsbestimmungen in Deutschland von 2011 bis 2022 um mehr als 28 % gestiegen [1]. Das verdeutlicht die Notwendigkeit eines zuverlässigen Systems zur Identifizierung und Schutz geistiger Werke.

1.1 Motivation

In der heutigen digitalisierten Welt, in der Informationen und Daten stetig mittels des Internets übertragen werden, gewinnt der Schutz geistigen Eigentums und die Sicherung der Authentizität von digitalen Inhalten eine immer größere Bedeutung. Ein immerwährender Wettlauf zwischen Kreativität und Piraterie hat dazu geführt, dass die Entwicklung robuster Sicherheitsmechanismen essenziell geworden ist. [2] Hierbei spielen digitale Wasserzeichen eine zentrale Rolle.

Digitale Wasserzeichen, inspiriert von den analogen Wasserzeichen in Papierdokumenten, sind unsichtbare oder schwer zu entfernende Kennzeichnungen, die in digitale Medien eingebettet werden, um deren Herkunft zu verifizieren und Integrität zu schützen. Ob in Bildern, Audioaufnahmen, Videos oder Dokumenten – die Anwendungsmöglichkeiten sind vielfältig und bieten einen effektiven Schutz gegen unerwünschte Manipulation und Plagiate durch Dritte. [3]

Der Einsatz digitaler Wasserzeichen eröffnet eine Vielzahl von Möglichkeiten für Künstler, Fotografen, Filmemacher und Autoren, ihre Werke vor unbefugter Verwendung und Verbreitung zu schützen und gleichzeitig die Sichtbarkeit ihrer Urheberschaft zu wahren. [3]

1.2 Zielstellung

Um digitale Bücher oder Artikel lesen zu können, muss oft ein bestimmtes Entgelt gezahlt werden. Um dies zu umgehen, könnte ein Angreifer auf die Idee kommen, diese Bücher oder Artikel herunterzuladen. Anschließend auf einer anderen Internetseite hochzuladen und weiteren Nutzern kostenfrei zur Verfügung zu stellen. Durch diese Möglichkeit würde der Verlag und/oder der Autor keinen Gewinn mehr machen. Aufgrund dessen bleibt die Frage offen, ob Verlage oder Autoren ihre Bücher bereits auf eine Art schützen, um Angriffe zu verhindern oder im Falle des oben genannten Szenarios den Angreifer ermitteln zu können. Wichtig sind diese ebenfalls, um Urhebernachweise in einem Strafverfahren finden zu können.

Ziel dieser Arbeit ist es, eine Analyse von digitaler Literatur, speziell von PDF-Dokumenten, durchzuführen, um mögliche Hinweise auf Identifikationsmerkmale (hier digitale Wasserzeichen) zu erhalten. Dabei sollen verschiedene Arten von Dokumenten untersucht und miteinander verglichen werden. Falls Unterschiede gefunden werden, soll außerdem untersucht werden, wo sich diese befinden, und wenn möglich, was diese aussagen.

Es gibt sechs Identifikationsmerkmale von digitaler Literatur, diese sind International Standard Book Number (ISBN), Digital Object Identifier (DOI), Uniform Resource Name (URN), Metadaten, Wasserzeichen und digitale Fingerabdrücke. Alle dienen dazu, digitale Werke eindeutig zu identifizieren, zu verwalten und zu schützen. Auf die ersten drei wird in dieser Arbeit nicht näher eingegangen. Auch die digitalen Fingerabdrücke werden lediglich als Synonym betrachtet, jedoch nicht näher erläutert.

1.3 Dokumentübersicht

Bevor mit der Betrachtung von digitalen Wasserzeichen begonnen werden kann, ist es sinnvoll, die Struktur einer PDF-Datei zu betrachten. Anschließend erfolgt ein erster Einblick in digitale Wasserzeichen. Dabei wird geklärt, was digitale Wasserzeichen überhaupt sind, welche Eigenschaften sie besitzen, wie ein Wasserzeichenalgorithmus grob funktioniert und wo digitale Wasserzeichen Anwendung finden.

In Kapitel 3 geht es genauer um die Möglichkeiten, die es gibt, ein digitales Wasserzeichen einzubinden. Dort werden verschiedene Wasserzeichenalgorithmen genauer beschrieben.

Da es immer Möglichkeiten gibt, vorhandene Schwachstellen auszunutzen werden in Kapitel 4 verschiedene Angriffsszenarien vorgestellt. Es wird geklärt, welche Art von Angriffen es auf ein digitales Wasserzeichen gibt und wie diese funktionieren.

Um Dokumente richtig zu untersuchen, müssen Vorüberlegungen getroffen und die Variationen untereinander abgewägt werden. Die Vorstellung und eine Diskussion, welche Variante genutzt werden soll, findet in Kapitel 5 statt.

In Kapitel 6 wird die Anwendung und Untersuchung von Dokumenten vorgestellt. Wichtig sind dabei die Vorbereitungen, die Durchführung und die anschließende Auswertung.

Kapitel 7 beantwortet zusammengefasst die Fragestellung, welche Kern dieser Arbeit ist.

2 Grundlagen

Wasserzeichen werden verwendet, um die Urheberrechte von beispielsweise Bild- und Videodateien zu schützen. Die Entwicklung der Technik und die Digitalisierung führten zu der Forschung und Entwicklung von digitalen Wasserzeichen Anfang der 90er Jahre [4]. Bei digitalen Wasserzeichen handelt es sich um „transparente, nicht wahrnehmbare Muster, welche in das Datenmaterial mit einem Einbettungsalgorithmus eingebracht werden“ [5].

Dabei gibt es verschiedene Arten, die unterschiedliche Eigenschaften und Anwendungsgebiete besitzen. Diese Beschreibungen zu einem Wasserzeichen sowie die Funktionsweise eines digitalen Wasserzeichens werden in den folgenden Unterkapiteln genauer vorgestellt. Da das Augenmerk auf PDF-Dokumenten liegt, wird vorab kurz die Struktur dieser erklärt.

2.1 Struktur einer PDF-Datei

Es ist eine Grundvoraussetzung, für die folgende Arbeit den Aufbau einer Portable Document Format (PDF)-Datei zu kennen. Mit diesen Kenntnissen sollen später aufgefundene Unterschiede besser eingeordnet werden.

Jede PDF-Datei besitzt eine klare Grundstruktur. Am Anfang ist der *Header* (Kopf) zu finden, welcher Informationen über die Version der PDF-Datei liefert. Der Header beginnt immer mit einem *%PDF*, auf den anschließend die Versionsnummer folgt, zum Beispiel %PDF-1.7. Die Versionsnummer gibt an, welcher PDF-Standard für die Erstellung der Datei verwendet wurde. Jede Version kann neue Funktionen, Verbesserungen und Änderungen am Format selbst einführen. Dies kann Einfluss auf die korrekte Darstellung und Verarbeitung der Dateien haben. [6]

Nach dem Header folgt der *Body* (Körper). Dieser ist in verschiedene Objekte unterteilt. Die einzelnen Unterobjekte beinhalten Metadaten des Dokumentes, Verweise für den Interpreter, Grundeinstellungen für die Seitengröße, Schrifteinstellungen und den eigentlichen Inhalt der Datei wie Text, Bilder und andere Medien. Es ist somit konkret das abgelegt, was dem Benutzer angezeigt wird. [6]

Im Anschluss an den Body ist eine Referenztabelle zu finden. In dieser Tabelle hat jedes Objekt der Datei einen Eintrag mit seiner jeweiligen Position. Diese Einträge betragen immer 20 Bytes. Eingeleitet wird die Tabelle mit einer Zeile, in der das Wort *xref* zu finden ist, siehe Abbildung 2.1. Darauf folgt eine Zeile mit zwei Zahlen. Die erste Zahl (z. B. 0) steht für die Objektnummer und die zweite Zahl (z. B. 484) für die Anzahl der Tabelleneinträge. Die folgenden Zeilen stellen die Tabelleneinträge dar. Dabei ist vor allem die erste Zahl wichtig, welche die Position des Objektes aufzeigt. [6]

```
xref
0 484
0000000000 65535 f
0000000015 00000 n
0000000439 00000 n
0000028668 00000 n
0000634371 00000 n
0000032436 00000 n
0000032079 00000 n
```

Abbildung 2.1: Beispiel Referenztabelle [eigene Abbildung]

Den Abschluss einer Datei bildet der Trailer, welcher beispielhaft in Abbildung 2.2 dargestellt ist. Dieser wird mit dem Wort *trailer* in einer einzelnen Zeile eingeleitet. Im Anschluss daran sind folgende Informationen gelistet:

- /Root: verweist auf das Katalogobjekt
- /Info: verweist auf das Objekt mit den Metadaten
- /Size: Anzahl der Einträge in der Referenztabelle

Nach diesen Informationen befindet sich eine Zeile mit *startxref*, gefolgt von einer Zeile, welche die Position der Referenztabelle in Bytes angibt. Die letzte Zeile bildet die Zeichenfolge *%%EOF*. Damit wird das Ende der Datei markiert. [6]

```
trailer
<</Root 363 0 R/ID [<cb62a3580e02814f3c868d18df7d9b28><1cc9bb160e8cb9ae8a1b2f8265daa579>]/Info 1 0 R/
Size 484>>
%iText-5.3.5
startxref
918428
%%EOF
```

Abbildung 2.2: Beispiel Trailer [eigene Abbildung]

2.2 Arten von digitalen Wasserzeichen

Bei Wasserzeichen wird zwischen verschiedenen Arten unterschieden. Es gibt die sichtbaren und unsichtbaren Wasserzeichen, wobei Letztere wiederum in robuste und fragile Wasserzeichen unterteilt werden können. Der Unterschied zwischen den Arten wird folgend erklärt.

2.2.1 Sichtbare Wasserzeichen

Innerhalb von digitalen Daten beinhalten sichtbare Wasserzeichen mehr oder weniger störende Fremdbildabschnitte, zum Beispiel das Symbol des Fernsehsenders in einer Bildecke. Durch diese Bildveränderung soll die kommerzielle Weiternutzung unmöglich gemacht werden, denn die Entfernung des Wasserzeichen ist nur schwer bis gar nicht möglich. [7]

Abbildung 2.3 zeigt zum Beispiel ein Magnolienbild, in welches sichtbar das Wort „Wasserzeichen“ eingefügt wurde. Die Entfernung des Wortes würde eine sichtbare Veränderung am Bild mit sich bringen. Dadurch wird eine unerwünschte Verbreitung verhindert und der Urheber kann sein Originalbild schützen, verwenden und verkaufen.

Abbildung 2.3: sichtbares Wasserzeichen [eigene Darstellung]

2.2.2 Unsichtbare Wasserzeichen

Im Vergleich zu den sichtbaren Wasserzeichen sind die unsichtbaren mit dem bloßen Auge nicht erkennbar. Diese kommen zum Einsatz, wenn die Anwendung der sichtbaren Wasserzeichen endet. So sollen sie beispielsweise nach Verkauf eines Bildes oder Veröffentlichung eines Artikels zum weiteren Schutz der Urheberrechte dienen. Dadurch kann später der Ursprung oder Nutzer eines Dokumentes / Bildes ermittelt werden. [8]

In der folgenden Abbildung 2.4 ist das gleiche Bild wie bei Abbildung 2.3 abgebildet, jedoch wurde diesmal ein unsichtbares digitales Wasserzeichen integriert.

Abbildung 2.4: unsichtbares Wasserzeichen [eigene Darstellung]

Der Nachweis, dass sich ein Wasserzeichen im Bild befindet, ist in Abbildung 2.5 zu erkennen. Darin sind Informationen zu dem Bild selbst sowie zu den eingebetteten Informationen enthalten. So kann abgelesen werden, dass es sich bei dem eingebetteten Dokument um eine Datei mit der Endung Rich Text Format (rtf) und einer Größe von 423 Byte handelt.

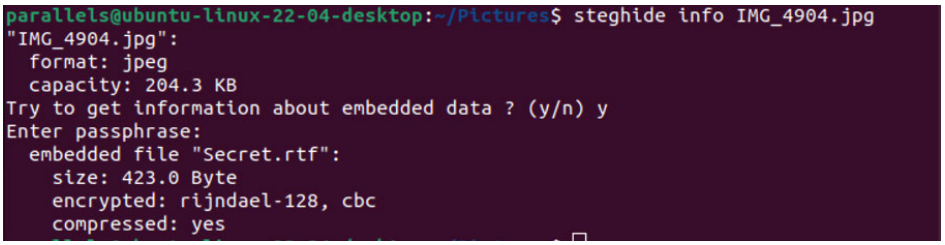

Abbildung 2.5: Nachweis Wasserzeichen [eigene Abbildung]

Robuste Wasserzeichen
Robuste Wasserzeichen sind eine effektive Methode zur Urheberidentifizierung und zum Urhebernachweis in digitalen Medien. Sie werden häufig eingesetzt, um digitale Inhalte wie Bilder, Videos oder Audiodateien mit unsichtbaren Markierungen zu versehen, die es ermöglichen, den ursprünglichen Urheber eines Werkes zu identifizieren. [9]

Der Hauptvorteil robuster Wasserzeichen liegt in ihrer Widerstandsfähigkeit gegen verschiedene Angriffe, die darauf abzielen, das Wasserzeichen zu entfernen oder zu manipulieren. Ein solches Wasserzeichen ist in der Lage, lange Zeit zu bestehen und somit eine zuverlässige Nachverfolgung des Urhebers zu gewährleisten. [10]

Es gibt zwei Haupttypen von Angriffen, denen robuste Wasserzeichen ausgesetzt sein können: Löschangriffe und Doppelmarkierungen.

Löschangriffe zielen darauf ab, das Wasserzeichen aus dem digitalen Medium zu entfernen, um die Identifizierung des Urhebers zu verhindern. Diese Angriffe können beispielsweise durch Bildbearbeitungstechniken oder Komprimierungsalgorithmen erfolgen. Trotz solcher Versuche ist ein robustes Wasserzeichen in der Lage, seine Integrität zu wahren und die Identifizierung des Urhebers auch nach solchen Manipulationen zu ermöglichen.

Doppelmarkierungen treten auf, wenn ein Angreifer versucht, ein bereits vorhandenes Wasserzeichen zu manipulieren, indem er ein weiteres Wasserzeichen darüberlegt. Das Ziel solcher Angriffe besteht darin, die ursprüngliche Markierung zu verdecken oder zu verwischen. Robuste Wasserzeichen sind jedoch darauf ausgelegt, solche Manipulationen zu erkennen und die ursprüngliche Markierung wiederherzustellen. Dadurch wird gewährleistet, dass der Urheber weiterhin identifiziert werden kann, selbst wenn das Wasserzeichen verändert wurde. [11] Diese und weitere Angriffe werden im Kapitel 4 näher beschrieben.

Fragile Wasserzeichen
Fragile Wasserzeichen stellen die schwächste Schutzvariante bei der digitalen Verhinderung von Bildfälschungen dar. Ihr Hauptzweck besteht darin, Veränderungen an einem Bild zu erkennen und zu lokalisieren. Im Gegensatz zu robusten Wasserzeichen, die gegen verschiedene Manipulationen widerstandsfähig sind, können fragile Wasserzeichen leicht entfernt werden und sind somit äußerst anfällig für Veränderungen. [12]

Die Fragilität des Wasserzeichens liegt darin, dass es bei kleinsten Veränderungen zerstört oder unlesbar wird. Dies kann beispielsweise durch geringfügige Farbveränderungen, Rauschen oder Entfernen eines Pixels ausgelöst werden. Sobald das Wasserzeichen beschädigt ist, kann es nicht wiederhergestellt werden. Daher kann ein fragiles Wasserzeichen genutzt werden, um eine Manipulation zu erkennen, jedoch nicht, um sie zu verhindern. [12, 13]

2.3 Eigenschaften von Wasserzeichen

Wasserzeichentechniken beschreiben die Art und Weise Informationen direkt in das Datenmaterial zu integrieren [14]. Diese erfordern verschiedene Eigenschaften, welche in den folgenden Abschnitten genannt und kurz beschrieben werden.

Robustheit
Eine Wasserzeicheninformation gilt als robust, sofern „die Informationen zuverlässig aus dem Datenmaterial ausgelesen werden können, auch wenn das Datenmaterial modifiziert, aber nicht vollständig zerstört, wurde." Demnach wird damit die Widerstandsfähigkeit der Wasserzeicheninformation gegenüber zufälligen Veränderungen beschrieben. Diese Veränderungen können sowohl absichtlich als auch unabsichtlich passieren. [10]

Kapazität
Hierbei wird angegeben, wie viele Informationen in die Originaldatei eingebracht werden können und wie viele Wasserzeichen gleichzeitig möglich sind. [15]

Nicht-Wahrnehmbarkeit
Bei dieser Eigenschaft wird sich auf das menschliche Wahrnehmungssystem bezogen. Eine Information gilt als „nicht-wahrnehmbar", wenn ein durchschnittliches Seh- und Hörvermögen zwischen der originalen Datei und dem markierten Datenmaterial nicht unterscheiden kann. [16]

Komplexität
Die Komplexität eines Wasserzeichenverfahrens beschreibt den Aufwand, der benötigt wird, um die Informationen einzubringen und wieder auszulesen. Weiterhin wird dargestellt, ob das Originalbild zum Auslesen verwendet werden muss oder nicht. [15]

Sicherheit
Diese Eigenschaft gibt an, wie sicher ein Wasserzeichenalgorithmus ist. Die Funktionsweise eines Wasserzeichenalgorithmus ist in Abschnitt 2.4 näher erläutert. Wie sicher ein Algorithmus ist, kann daran gemessen werden, wie resistent er beispielsweise gegen Zerstörung, Änderung, Fälschung oder Aufspürung ist. Dies wird jedoch unter den Bedingungen gemessen, dass der Angreifer Wissen über das Wasserzeichenverfahren sowie mindestens ein markiertes Datenmaterial besitzt. Zusammengefasst wird damit die „Sicherheit gegen gezielte (nicht-blinde) Angriffe" beschrieben. [16]

Geheime / öffentliche Verifikation
Die Eigenschaft „geheime / öffentliche Verifikation" gibt an, wer die Möglichkeit hat das Wasserzeichen aufzudecken. Unterschieden wird dabei zwischen Urheber, spezieller Personen-

gruppe oder Allgemeinheit. Es ist jedoch schwierig ein sicheres, öffentliches Wasserzeichen zu erzeugen, da der verwendete Schlüssel benötigt wird. Sobald dieses öffentlich verfügbar und für jeden zugänglich ist, kann das Wasserzeichen einfach entfernt werden. [15]

Konkurrenz der Eigenschaften
Die zuvor beschriebenen Eigenschaften konkurrieren miteinander und können somit nicht zeitgleich optimiert werden. Wird eine große Menge an Informationen benötigt, kann nicht gleichzeitig die Nicht-Wahrnehmbarkeit und Robustheit optimiert werden. [17]

Die grundsätzlichen Anforderungen an Wasserzeichenverfahren bilden Kapazität, Robustheit und Nicht-Wahrnehmbarkeit. Dabei muss beachtet werden, dass sie in einer Art Gleichgewicht stehen und dadurch nicht gleichzeitig beliebig verändert werden können. Zur Veranschaulichung können die Eigenschaften zusammen als ein Quader dargestellt werden, bei dem die Achsen die Eigenschaften bilden und das Volumen immer konstant ist (siehe Abbildung 2.6). Wird eine Eigenschaft erhöht, so muss sich automatisch eine andere Eigenschaft verringern, damit das Volumen konstant bleibt. [18] Im dargestellten Beispiel wird die Eigenschaft „Nicht-Wahrnehmbarkeit“ erhöht, wodurch sich die Eigenschaft „Robustheit “ verringert.

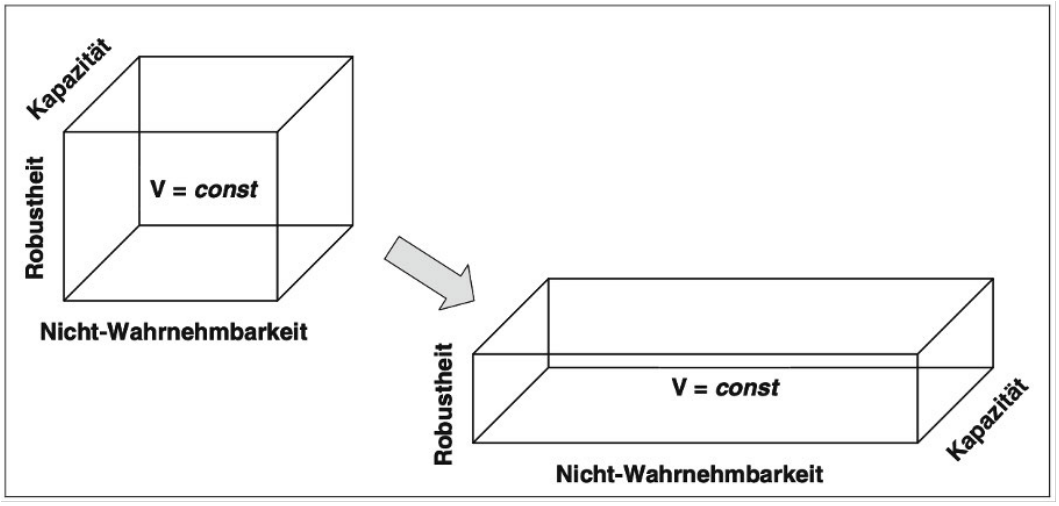

Abbildung 2.6: Zusammenhang der Eigenschaften von Wasserzeichen [19]

2.4 Funktionsweise eines Wasserzeichenalgorithmus

Ein Wasserzeichenalgorithmus ist ein Verfahren, das verwendet wird, um digitale Wasserzeichen in Medien wie Dokumenten, Bildern, Videos oder Audiodateien einzubetten. Das Ziel eines Wasserzeichens ist es, die Integrität und Authentizität des Mediums zu schützen, indem es unsichtbar und schwer zu entfernen ist. [20]

Es folgt eine grundlegende Erklärung, wie ein Wasserzeichenalgorithmus funktionieren kann:

1. **Codierung des Wasserzeichens:** Das Wasserzeichen wird in eine Form umgewandelt, die in das Medium eingebettet werden kann. Es kann sich um eine Sequenz von Bits, eine digitale Signatur oder andere Informationen handeln, die auf das Medium angewendet werden sollen. Innerhalb des Wasserzeichens können Informationen über den Urheber oder Besitzer des Mediums eingebettet sein. [21]
2. **Auswahl der Einbettungsstelle:** Der Algorithmus wählt strategisch die Stellen im Medium aus, an denen das Wasserzeichen eingefügt werden soll. Diese sollten so ausgewählt werden, dass das Wasserzeichen sowohl robust als auch schwer zu erkennen ist. Das kann zum Beispiel durch die Ausnutzung von Schwachstellen in der menschlichen Wahrnehmung erreicht werden. [21]
3. **Einbettung des Wasserzeichens:** Das Wasserzeichen wird in das Medium eingefügt, indem es mit den vorhandenen Daten kombiniert wird. Dies kann durch das Hinzufügen von Bits in das Binärbild, das Modifizieren der Farbwerte oder das Verändern von Frequenzen erfolgen. Der Algorithmus stellt dabei sicher, dass das eingebettete Wasserzeichen das ursprüngliche Medium nicht erkennbar verändert. [21]
4. **Überprüfung des Wasserzeichens:** Nachdem das Wasserzeichen eingebettet wurde, kann es durch den Algorithmus überprüft werden, um die Integrität des Mediums zu bestätigen. Dazu wird das Medium analysiert, damit das Wasserzeichen extrahiert und mit dem ursprünglichen verglichen werden kann. Wenn das Wasserzeichen erfolgreich extrahiert werden kann und mit dem Original übereinstimmt, wird die Echtheit des Mediums bestätigt. [21]

Es ist wichtig zu beachten, dass Wasserzeichenalgorithmen keine absolute Sicherheit bieten können. Mit ausreichendem Aufwand und spezifischem Fachwissen ist es möglich, Wasserzeichen zu entfernen oder zu manipulieren. Dennoch sind sie ein nützliches Instrument, um die Urheberschaft zu schützen und die Integrität digitaler Medien zu wahren.

2.5 Anwendungsgebiete von Wasserzeichen

Die Anwendungsgebiete von digitalen Wasserzeichen sind zahlreich. In all diesen Bereichen sorgen digitale Wasserzeichen für eine höhere Sicherheit. [22]

Im Folgenden werden einige ausgewählte Anwendungsgebiete vorgestellt:

Urheberschaftsnachweis
In das Datenmaterial wird von Autoren, Urhebern oder Produzenten eine eindeutige Markierung eingefügt. Bei dieser handelt es sich um Informationen über die Eigentumsrechte. Im Anschluss wird das veränderte Datenmaterial verbreitet und der Urheber behält das Original. Dadurch können immer Rückschlüsse auf den Urheber gezogen werden. [23]

Beispielsweise erzeugt der Urheber eines Bildes ein digitales Wasserzeichen und legt dieses anschließend über das Bild, welches jedoch nicht sichtbar ist und somit auch bei Verkauf des Bildes nicht entfernt werden muss. Dadurch lässt sich später der Urheber nachweisen.

Integritätsnachweis
Die eingebrachten Informationen erlauben die Feststellung, ob es sich bei dem Dokument um das Original handelt oder ob das Datenmaterial manipuliert wurde. Es handelt sich somit um den Nachweis der Unversehrtheit, denn die Informationen beinhalten Sicherheitsinformationen, zum Beispiel Prüfsumme oder Zeitstempel. Weiterhin spiegelt die Wasserzeicheninformation die Semantik, also die Bedeutung des Datenmaterials, wider. [23]

Kundenidentifizierung oder Transaktionskontrolle
Hierbei wird die Authentifizierung thematisiert. So wird ein kundenspezifisches Merkmal, zum Beispiel die Kunden-ID, in das Datenmaterial eingebunden, um nach der Verbreitung legale Kundenkopien erkennen zu können. Gleichzeitig soll es dazu dienen, dass im Fall von illegalen Kopien sich diese zum Erzeuger zurück verfolgen lassen. [23]

Kopierschutz und Übertragungskontrolle
Bei diesem Anwendungsgebiet kann für jedes Dokument festgelegt werden, wie es weiter verwendet werden darf. Es werden also Zugriffs- und Verwendungsinformationen gespeichert. So kann beispielsweise der Autor eines Artikels festlegen, dass der Artikel zwar gelesen, jedoch nicht beschrieben werden darf. Beim Kopierschutz geht es darum, dass nach einer Kopie das Wasserzeichen noch vorhanden ist und diese somit nicht einfach vervielfältigt und verbreitet werden kann. [22]

3 Algorithmen zur Erstellung von digitalen Wasserzeichen

Es gibt verschiedene Algorithmen zur Erstellung digitaler Wasserzeichen, die je nach Anwendungsgebiet und gewünschter Eigenschaften verwendet werden können. Es ist wichtig zu beachten, dass die Wahl des geeigneten Algorithmus von verschiedenen Faktoren abhängt, wie zum Beispiel der Art des zu schützenden Inhalts, den gewünschten Robustheitseigenschaften und den potenziellen Bedrohungen oder Angriffsszenarien. Dabei kann zwischen den folgenden zwei Wasserzeichenverfahren unterschieden werden:

Nicht-blinde Wasserzeichenverfahren: Diese Art von Wasserzeichenverfahren benötigen das Originalbild, um das Wasserzeichen auslesen zu können. Weiterhin wird das verwendete Wasserzeichen sichtbar dargestellt. [16]

Blinde Wasserzeichenverfahren: Das ist eine Technik, bei der ein digitales Wasserzeichen in eine Datei eingebettet wird, ohne dass der ursprüngliche Inhalt dafür benötigt wird. Dies ermöglicht die nachträgliche Überprüfung der Authentizität oder Herkunft der Datei, ohne den Originalinhalt offenzulegen. Das Originalbild wird zum Auslesen also nicht benötigt. [16]

Nachstehend werden einige gängige Algorithmen näher erläutert.

Bei der Einbettung von Wasserzeichen kann zwischen dem räumlichen Bereich und dem transformierten Bereich unterschieden werden. Bei der räumlichen Einbettung wird in den Wert jeden Pixels das Wasserzeichensignal eingebunden. Hingegen im Frequenzbereich geschieht die Einbettung in die Koeffizienten des transformierten Bildes. Das bedeutet, dass eine bestimmte Transformation (siehe Abschnitt 3.2) durchgeführt wird und anschließend die Werte aus dem geänderten Bild genommen werden, um das Wasserzeichen einzubetten. In diesem Bereich sind die Ansätze robuster gegenüber Rauschen und Angriffen. [24]

3.1 Spread Spectrum Watermarking

Bei dem Spread Spectrum Watermarking (SSW) handelt es sich um einen Algorithmus, der die Frequenzbereiche eines Bildes nutzt. Wichtig ist, dass das Wasserzeichen nicht in wahrnehmungsmäßig unbedeutenden Stellen im Bild platziert wird, da diese durch Signal- und Geometrieprozesse beeinflusst werden können. Der Frequenzbereich eines Bildes wird als Kommunikationskanal betrachtet und somit das Wasserzeichen als Signal, welches darüber übertragen wird. [24]

Bei der Spread Spectrum Kommunikation wird über eine größere Bandbreite ein Schmalbandsignal übertragen. Als Folge kann die vorhandene Signalenergie nicht nachgewiesen werden. Ähnlich dazu „wird das Wasserzeichen über viele Frequenzbereiche verteilt, sodass die Energie in einem einzelnen Bereich sehr gering und somit nicht nachweisbar ist“. [24]

Innerhalb der Spread Spectrum Techniken gibt es zwei Arten:

1. Direct Sequence Spread Spectrum (DS-SS)
2. Frequency Hopping Spread Spectrum (FH-SS)

Bei dem DS-SS kann ein schwaches Breitbandsignal leicht im gleichen Spektrum wie ein starkes Signal versteckt werden. Ist dies der Fall, so erscheint jedem Signal das jeweils andere als Rauschen. Die Hauptkomponente beider Techniken ist eine Pseudo Random Noise Sequence (PRNS) (dt. Pseudo-Zufallsrauschsequenz). Innerhalb des DS-SS wird der ursprüngliche Basisbitstrom mit dieser PRNS multipliziert, wodurch ein neuer Bitstrom erzeugt wird. Im Nachhinein können nur Empfänger mit der richtigen PRNS das Originalbild herstellen. [24]

Beim FH-SS Algorithmus ist ein periodischer Wechsel der Übertragungsfrequenz in Verwendung. Das Frequenzsignal wird als eine Folge von modulierten Datenbursts mit zeitlich veränderlichen, pseudozufälligen Trägerfrequenzen betrachtet. Datenbursts bezeichnen eine Übertragung mit relativ hoher Bandbreite über einen kurzen Zeitraum [25]. „Die Menge der möglichen Trägerfrequenzen wird als Hopset bezeichnet“ [24]. Hopping bezeichnet eine Technik, bei der das Wasserzeichen-Signal während der Einbettung über verschiedene Frequenzen oder Zeitbereiche „springt“ [26]. Dieses wird über eine Anzahl von Kanälen durchgeführt, bei dem jeder Kanal als Spektralbereich mit einer zentralen Frequenz im Hopset definiert wird. [24]

Wasserzeichen Einbettung
Innerhalb der Einbettung wird das DS-SS verwendet, um dem Signal mehr Robustheit zu geben und das FH-SS, um die Position der Einbettung zu bestimmen. [24]

Zuerst wird eine Informationsbitfolge aus -1 und 1 durch Multiplikation mit einem großen Faktor (Chip-Rate C_r) gespreizt. Die Größe der Sequenz ist gleich zu der Multiplikation von Chip-Rate mit der Anzahl der Informationsbits. Um das Wasserzeichensignal zu erhalten, wird die gespreizte Sequenz mit einer binären Pseudozufallsrauschsequenz moduliert. Diese wird anschließend mit einem lokal einstellbaren Amplitudenfaktor verstärkt (siehe Abbildung 3.1 Block A). [24]

Danach folgt die Einbettung des Wasserzeichens. Dafür wird ein geheimer Schlüssel, eine zufällig generierte Position, das Originalbild sowie das Ergebnis aus Block A benötigt. Das Ergebnis ist ein mit einem Wasserzeichen markiertes Bild (siehe Abbildung 3.1). [24]

1. Beispiel (Bilder)
Wird ein Bild mit der Größe 256 x 256 Pixel betrachtet, so sind 65.536 Pixel verfügbar. Diese werden als Hopset betrachtet. Für die Einbettung werden 10 % und somit 6.553 Stellen benötigt. Diese werden pseudozufällig ohne Wiederholung bestimmt. Ein Wasserzeichen-Bit wird durch Addition in ein Pixel eingebettet. Die Ausgabe stellen Pixel mit Wasserzeichen dar. [24]

Für die Wiederherstellung der eingebetteten Information müssen die Einbettungsstellen genau bestimmt werden. Pixel, die das Wasserzeichen enthalten, werden mit der Pseudozufallsrauschsequenz von der Generierung korreliert. Die Korrelation beinhaltet eine Demodulation und anschließende Addition. Demodulation ist der Prozess der Umwandlung eines modulier-

ten Signals (z. B. eines Radiosignals) zurück in seine ursprüngliche Form, um die übertragene Information wiederherzustellen. Abschließend wird durch das Vorzeichen der Korrelationssumme das Informationsbit bestimmt. [24]

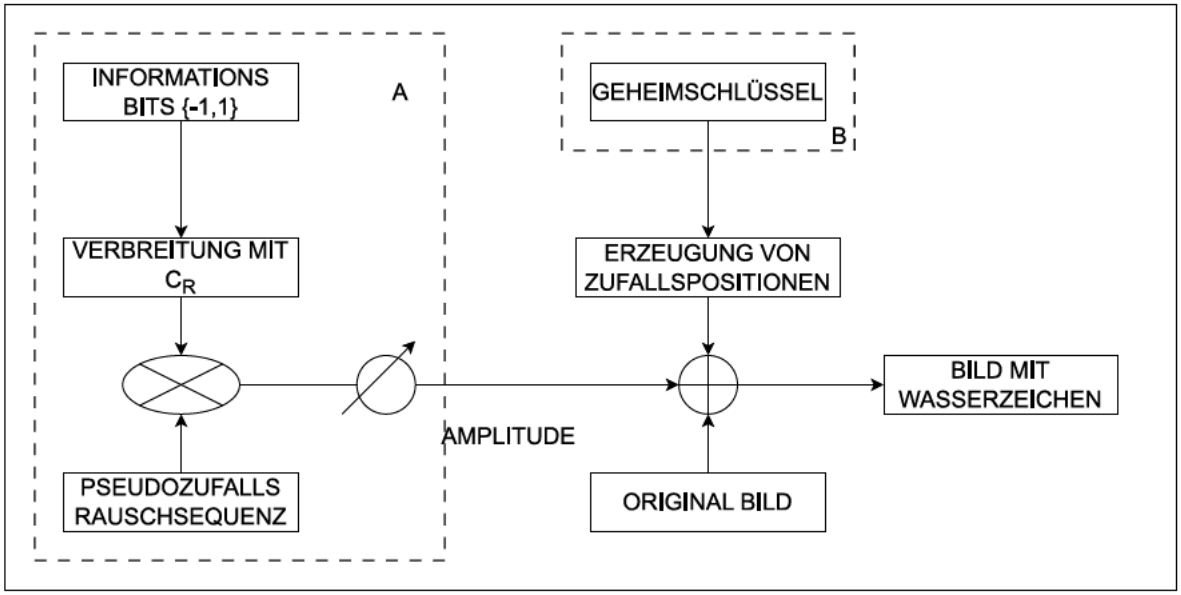

Abbildung 3.1: Einbettung Spread Spectrum Watermarking (adaptiert nach [24])

2. Beispiel (PDF-Dokumente)
Zuerst müssen folgende Annahmen getroffen werden:

- Es gibt ein Original-PDF-Dokument, welches mit einem unsichtbaren Wasserzeichen geschützt werden soll.
- Das Wasserzeichen besteht aus einer binären Sequenz von 0 und 1.
- Bei der Verwendung der Spread Spectrum Methode, wird jedes Bit des Wasserzeichens auf eine bestimmte Frequenzkomponente des Dokuments abgebildet.

Anschließend werden drei Schritte durchgeführt, um das Wasserzeichen in ein PDF-Dokument einzubetten. Als erstes muss das PDF-Dokument in den Frequenzbereich konvertiert werden. Dies kann mithilfe der diskreten Fourier-Transformation (DFT) oder der diskreten Wavelet-Transformation (DWT) (siehe Abschnitt 3.2) erfolgen. Der nächste Schritt ist die Einbettung des Wasserzeichens. Dabei wird jedes Bit des Wasserzeichens auf eine bestimmte Frequenzkomponente abgebildet. Das kann durch Ändern des Amplitudenwertes der jeweiligen Frequenzkomponente erfolgen. Zum Beispiel könnte das Muster 0 zu einer leichten Verringerung der Amplitude führen und das Muster 1 zu einer leichten Erhöhung. Dieser Prozess wird für alle Bits des Wasserzeichens wiederholt. Der letzte Schritt ist die Rücktransformation. Dabei wird die Inverse, der zuvor durchgeführten Transformation gebildet, um das modifizierte Frequenzsignal zurückzubringen und das wasserzeichenhaltige Dokument zu erhalten.

3.2 Diskrete Wavelet-Transformation

Die diskrete Wavelet-Transformation (DWT) ist ein mathematisches Verfahren zur Analyse von Signalen und Bildern. Die Grundidee dieser Transformation liegt in der Trennung von Frequenzdetails. Dabei handelt es sich um eine Zerlegung mit mehreren Auflösungen. In Abbildung 3.2 ist zu sehen, dass das Hauptbild in vier gleich große Unterbilder zerlegt wird. Eines

der Unterbilder repräsentiert die niedrigen Frequenzen des Bildes. Es enthält Informationen über die groben Strukturen und allgemeinen Eigenschaften des Bildes. Diese Untergrafik wird oft als Approximation oder Niederfrequenzkomponente bezeichnet. Die anderen drei Unterbilder repräsentieren die hohen Frequenzen des Bildes. Diese Unterbilder enthalten Informationen über feinere Details wie Bildkanten, Konturen und Texturen. Jedes der Unterbilder für hohe Frequenzen ist auf eine bestimmte Art von Strukturen spezialisiert, zum Beispiel horizontale, vertikale oder diagonale Kanten. [27]

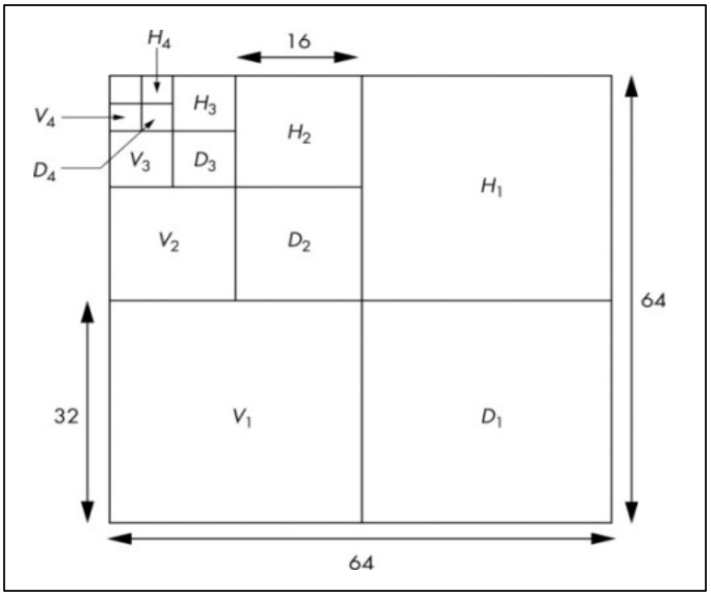

Abbildung 3.2: Diskrete Wavelet Transformation [27]

Innerhalb der hohen Frequenzen kann die Einbettung nicht leicht erkannt werden, da sie für den Menschen schwer wahrnehmbar ist. Nach einem Angriff auf diese Bereiche ist die Stabilität gering. [27]

In Bildern ist die meiste Energie in den niedrigen Frequenzen konzentriert, sodass die hochfrequenten Teile weniger energiereich sind. Daher kann die Wavelet-Transformation verwendet werden, um eine energieeffiziente Darstellung des Bildes zu erhalten, indem die hochfrequenten Teile komprimiert oder vernachlässigt werden. Niedrige Frequenzkoeffizienten sind gegenüber üblichen Angriffen nahezu unverändert, sodass diese eine bessere Robustheit aufweisen. [27]

Die Wavelet-Transformation kann in mehreren Stufen durchgeführt werden, um eine detailliertere Analyse des Bildes zu ermöglichen. In einer vierstufigen DWT (wie in Abbildung 3.2 zu erkennen ist) werden das Hauptbild und die Unterbilder für hohe Frequenzen weiter in jeweils vier neue Unterbilder zerlegt, bis die gewünschte Detailtiefe erreicht ist. Dadurch wird eine hierarchische Darstellung des Bildes erzeugt, die verschiedene Skalen und Frequenzbereiche umfasst. Die Wavelet-Transformation wird in verschiedenen Bereichen wie Bildverarbeitung, Signalverarbeitung, Datenkompression und Mustererkennung eingesetzt. Sie ermöglicht eine effektive Analyse und Darstellung von Signalen und Bildern sowohl in Zeit- als auch in Frequenzdomänen. [27]

3.3 Least Significant Bit Watermarking

Das einfachste Wasserzeichenverfahren für Bilder im räumlichen Bereich besteht darin, ein Wasserzeichen in die am wenigsten bedeutsamen Bits einiger zufällig ausgewählter Pixel des Bildes einzubetten. [28] Daher handelt es sich bei dem Least Significant Bit (LSB) Algorithmus um eine sehr häufig angewandte Methode.

Ein Pixel besteht aus einem Byte und somit aus acht Bits. Der Algorithmus beschreibt die Verwendung des niederwertigsten Bits (siehe Abbildung 3.3). Für die Wasserzeicheninformation wird das am weitesten rechts stehende Bit verwendet, wobei es sich um das am wenigsten bedeutsame Bit handelt. [29]

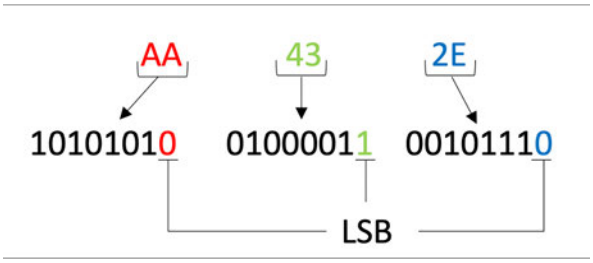

Abbildung 3.3: Least Significant Bit [eigene Abbildung]

Beim Basisalgorithmus wird nur ein Bit des Originalbildes durch den entsprechenden Wert des Wasserzeichens ersetzt. Es kann jedoch auch vorkommen, dass das Wasserzeichen in zwei Bits eingebettet wird. Dadurch bekommt es mehr Sicherheit. [28] Ebenso muss erwähnt werden, dass sobald ein Angreifer den Algorithmus kennt, dieser das Wasserzeichen ohne großen Aufwand verändern kann. [29]

Beispiel
Ein Pixel besteht aus RGB-Farbkomponenten. Durch den vorliegenden Algorithmus werden die RGB-Komponenten bei Änderung des LSB nicht wesentlich verändert.

Beispielsweise werden die Pixel (AA|43|2E), (EF|12|33) und (CD|59|B4) betrachtet, in welche der Buchstabe „S“ eingepflegt werden soll.

Binärwert des Zeichens S: 0101 0011

Tabelle 3.1: Beispiel Einbettung des Buchstabens S in drei Pixel

Pixel-komponente	**Binärwert**	**Zeichen „*S*“**	**Daten nach Einbettung**	**Ausgabe**
AA	10101010	0	1010101**0**	AA
43	01000011	1	0100001**1**	43
2E	00101110	0	0010111**0**	2E
EF	11101111	1	1110111**1**	EF
12	00010010	0	0001001**0**	12
33	00110011	0	0011001**0**	32
CD	11001101	1	1100110**1**	CD
59	01011001	1	0101100**1**	59
B4	10110100		10110100	B4

In diesem Beispiel ist deutlich zu erkennen, dass die Einbettung des Buchstabens nur eine Änderung bei dem Pixel mit dem Hexadezimalwert 33 aufweist. Die Änderung vom Wert 33 auf 32 ist jedoch so gering, dass es sich immernoch um die gleiche Farbe handelt und somit keine sichtbare Veränderung am Originalbild vorhanden ist. Aufgrund dessen lässt sich dieses Verfahren einfach und ohne große Veränderungen anwenden. [29]

4 Angriffe auf Wasserzeichen

Wasserzeichen sind ein großer Fortschritt, um den Urheber zu schützen und zurückverfolgen zu können. Dennoch gibt es auch hier Möglichkeiten, die dazu führen, dass ein Wasserzeichen verändert oder sogar komplett entfernt werden kann. Dabei kann zwischen zwei Angriffsarten unterschieden werden:

Freundliche Angriffe haben das Ziel, Änderungen am Datenmaterial vorzunehmen, jedoch das Wasserzeichen nicht zu zerstören. Dazu gehört die Nachbearbeitung an digitalen Medien, zum Beispiel die Konvertierung in ein anderes Format. [30]

Feindliche Angriffe hingehen haben das Ziel, das Wasserzeichen zu zerstören. Dazu gehören die Mehrfachmarkierung und die Zugabe von Rauschanteilen. [30]

In den folgenden Unterkapiteln werden einige Angriffe auf verschiedene Eigenschaften von Wasserzeichen genannt und näher vorgestellt.

4.1 Angriffe auf die Robustheit

Die Robustheit ist eine der wichtigsten Eigenschaften bei einem Wasserzeichenverfahren. Innerhalb der Angriffe geht es nicht darum, dass Wasserzeichen zu entfernen, sondern lediglich um Änderungen am Datenmaterial selbst. Dennoch gibt es eine Auswahl an Robustheitstests, die durchgeführt werden können. So stehen beispielsweise der StirMark-Angriff und die Mosaikattacke zur Verfügung. [11] Diese Möglichkeiten sollen im Folgenden näher erläutert und charakterisiert werden.

4.1.1 StirMark-Angriff

StirMark ist ein generisches Tool, das für einfache Robustheitstests von Bildmarkierungsalgorithmen und anderen steganografischen Techniken entwickelt wurde. In seiner einfachsten Version simuliert StirMark einen *Resampling*-Prozess (dt. Wiederholungsprüfung), das bedeutet, es fügt dieselbe Art von Fehlern in ein Bild ein, als würde es gedruckt und anschließend erneut eingescannt werden. [31]

Es wird somit eine Kombination aus geometrischen Transformationen, zum Beispiel Drehung, Stauchung oder Verzerrung, und Kompression durchgeführt. Bei einer Kompression handelt es sich um eine Größenänderung. [11]

Nach diesen Veränderungen kommt es zu einem *Resampling*, was mit einer erneuten Digitalisierung gleichzusetzen ist. Zusammengefasst kann die Simulation mit „Ausdrucken und neu Einscannen“ beschrieben werden. [11]

4.1.2 Mosaikattacke

Bei diesem Angriff wird das Originalbild in eine Reihe kleinerer Teilbilder zerlegt. Diese Teilbilder sind kleiner als die bei typischen Suchmaschinen zur Auffindung von Wasserzeichen festgelegte Minimalgröße. Dadurch wird das digitale Wasserzeichen durch die gezielte Platzierung von Mosaiken in den betroffenen Bildbereichen gestört. Ein Mosaik ist eine Musterung aus kleinen quadratischen oder rechteckigen Blöcken mit unterschiedlichen Farben. [32] Zusammengefasst zerlegt ein Angreifer das Bild in mehrere Einzelbilder und setzt diese anschließend wieder zusammen. [33]

In Abbildung 4.1 wurde das Originalbild (rechts) in sechs Einzelbilder (links) zerlegt. Beim Laden der Internetseite wird beim linken Bild jedes Einzelbild separat und nacheinander geladen. Hingegen das Originalbild pixelweise und somit zeilenweise geladen wird.

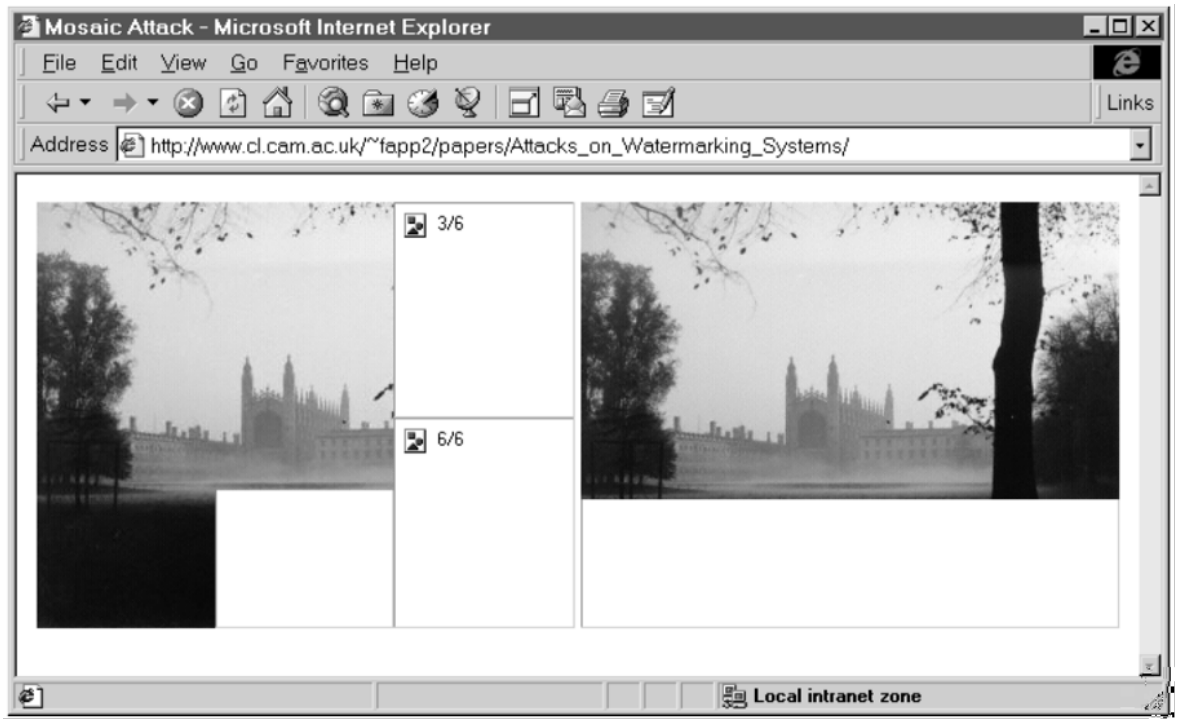

Abbildung 4.1: Mosaikattacke [32]

Die Mosaikattacke kann verschiedene Ziele haben. Ein häufiges Ziel ist es, das Wasserzeichen zu entfernen, um eine unlizenzierte Nutzung des geschützten, digitalen Inhalts zu ermöglichen. In einigen Fällen kann die Mosaikattacke auch verwendet werden, um ein fremdes Wasserzeichen durch ein anderes zu ersetzen, die Urheberschaft zu fälschen oder die Rückverfolgbarkeit des Inhalts zu verhindern. [32]

Um die Mosaikattacke zu bekämpfen, müssen spezielle Wasserzeichenalgorithmen entwickelt werden, die widerstandsfähig gegen solche Angriffe sind. Dies kann durch die Verwendung robuster Wasserzeichenverfahren erreicht werden, welche die Mustererkennung in Bildern nutzen, um das Wasserzeichen auch bei Vorhandensein von Mosaiken wiederherzustellen. Zusätzlich können Techniken wie Verschlüsselung und steganographische Verfahren angewendet werden, um die Sicherheit des Wasserzeichens zu verbessern und Manipulationsversuche zu erschweren. [32]

Der ursprüngliche Grund für die Entwicklung der Mosaikattacke bestand darin, Web-Crawler, die nach illegal kopiertem Bildmaterial suchen, zu irritieren. Web-Crawler sind „Software-Programme, die das Internet durchsuchen. Dabei analysieren und indizieren sie Inhalte von Webseiten wie Texte und Bilder oder auch Videos." [34] Der Crawler findet nun, anstatt des gesamten Bildes, wie es der Betrachter sieht, die Einzelbilder. Diese sind jedoch so klein, dass darin kaum Informationen für die Urheberidentifizierung gespeichert werden können. [32]

Blinde Verfahren haben besonders große Schwierigkeiten Angriffen zu widerstehen, da es für diese schwierig ist, Angriffe festzustellen und anschließend darauf zu reagieren. [11]

4.2 Angriffe auf die Eindeutigkeit des Urhebers

Einige Wasserzeichenverfahren sind darauf ausgelegt mehrere Wasserzeichen aufzunehmen. Schwierig wird dies jedoch, sobald ein Angreifer seine eigene Urheberinformation in das Datenmaterial mit einbettet. Jedes Mal wenn das passiert, ist nicht eindeutig feststellbar, welches Wasserzeichen zuerst vorhanden war, denn sowohl Urheber als auch Angreifer können die Wasserzeicheninformation extrahieren. Deshalb wird dieser Angriff auch als **Doppelmarkierung** oder **Invertierbarkeitsproblem** bezeichnet. [33]

Abbildung 4.2 zeigt ein schematisches Beispiel für eine Doppelmarkierung durch den Urheber und einen Angreifer. In diesem Fall stellt Alice (Urheber) ihr Bild, welches mit einem nicht-blinden Wasserzeichenverfahren versehen ist, öffentlich zur Verfügung. Dieses Bild findet Bob (Angreifer) und erstellt dazu sein eigenes Wasserzeichen. Dies führt dazu, dass er ein neues Originalbild bekommt. Die Überprüfung beider Bilder mit ihrem Wasserzeichen führt zum selben Ergebnis, wodurch nicht feststellbar ist, welches Bild nun das echte Original ist, da beide Personen ein „Originalbild" besitzen. [33]

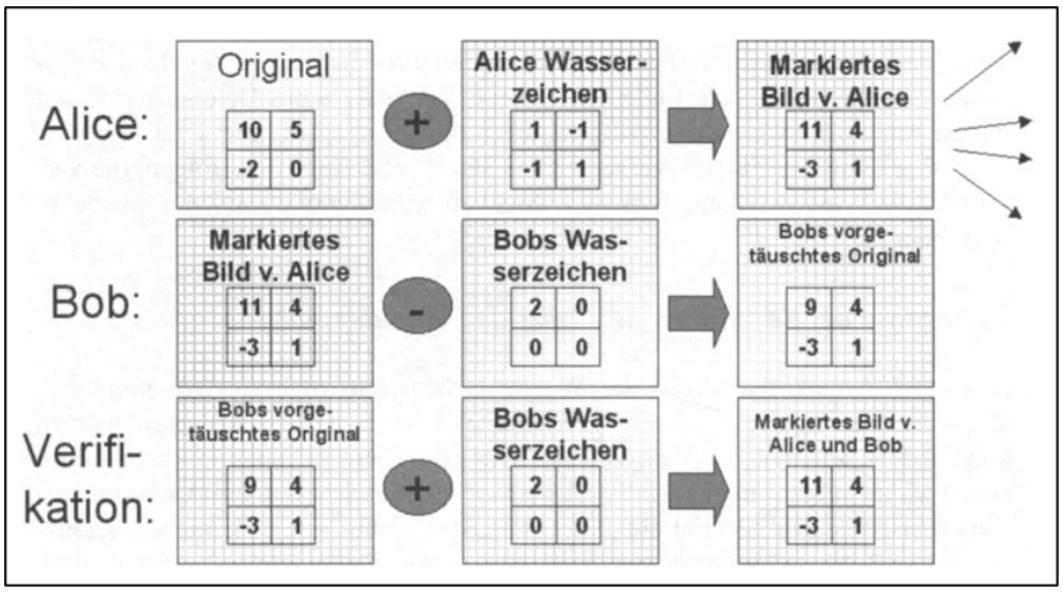

Abbildung 4.2: Doppelmarkierung durch Urheber und Angreifer [33]

Eine Lösung für eine Doppelmarkierung ist die Verwendung eines blinden Wasserzeichenverfahrens, sodass ein Angreifer nicht einfach sein Wasserzeichen darüber legen kann. Weiterhin können Zeitstempel verwendet werden, um im Nachhinein überprüfen zu können, welches Wasserzeichen zuerst eingebettet wurde. [33]

4.3 Angriffe auf das Wasserzeichen selbst

Bei diesem Angriff geht es um spezielle Änderungen im Datenmaterial, um dadurch das Wasserzeichen verändern zu können. Der Angriff ist ebenfalls bekannt unter dem Namen **Histogrammattacke**. [30]

Das Histogramm eines Bildes repräsentiert die Verteilung der Helligkeitswerte oder Farbwerte in einem Bild. [35] Einige Wasserzeichenverfahren basieren auf der Annahme, dass das Wasserzeichen die Histogrammverteilung des Originalbildes beeinflusst und somit als Signatur des Wasserzeichens verwendet werden kann. [33]

Bei einer Histogrammattacke versucht ein Angreifer, das Wasserzeichen zu schwächen oder zu entfernen, indem er gezielte Veränderungen am Histogramm des Bildes vornimmt. Dies kann zum Beispiel dadurch passieren, dass durch bestimmte Manipulationen an den Helligkeits- oder Farbwerten die charakteristischen Eigenschaften des Wasserzeichens zerstört oder verändert werden. [36]

Es gibt verschiedene Methoden, um eine Histogrammattacke durchzuführen. Einige Beispiele umfassen:

1. **Histogramm-Ausgleich:** Der Angreifer kann versuchen, den Kontrast des Bildes zu erhöhen oder zu verringern, um die Unterschiede zwischen den Wasserzeichen- und Nicht-Wasserzeichen-Bereichen zu reduzieren. [36]
2. **Histogramm-Skalierung:** Durch Anpassung der Skalierung des Histogramms kann der Angreifer versuchen, die Intensität der Wasserzeichenmerkmale zu verringern oder zu verstärken. [36]
3. **Histogramm-Manipulation:** Der Angreifer kann gezielt die Verteilung der Helligkeits- oder Farbwerte im Histogramm ändern, um die charakteristischen Eigenschaften des Wasserzeichens zu verändern oder zu eliminieren. [36]

Es ist wichtig zu beachten, dass die Effektivität einer Histogrammattacke von vielen Faktoren abhängt, einschließlich der spezifischen Wasserzeichenmethode, der Qualität des Wasserzeichens und der Fähigkeiten des Angreifers. [36]

4.4 Angriffe auf die Übertragbarkeit des Wasserzeichens auf andere Dokumente

Ziel des Angriffs ist es nicht, den Ausleseprozess oder die eingebetteten Informationen zu stören oder zu zerstören. Wichtiger ist das Konzept der Wasserzeichen Anwendung, so dass dieses angegriffen wird. Innerhalb des Angriffs sollen Schwachstellen oder Fehler eines Pro-

gramms aufgedeckt werden, damit diese anschließend als „Hintertür“ ausgenutzt werden können. Allgemein lässt sich sagen, dass es relevant ist, die Anwendung zu hintergehen und irrelevant den verwendeten Algorithmus herauszufinden. [37]

Zusammengefasst handelt es sich bei dieser Angriffsart um eine Kopie des Wasserzeichens, ohne das Wasserzeichenverfahren oder den Wasserzeichenschlüssel zu kennen. Sobald diese Kopie auf andere Medien übertragen wird, können bestimmte Leistungen in Anspruch genommen werden, die ohne Wasserzeichen nicht möglich wären. Zum Beispiel kann das Wasserzeichen eines Videos auf eine Raubkopie übertragen werden, um es anschließend abspielen zu können. [33]

4.5 Angriffe auf unterschiedliche Kopien

Hierbei handelt es sich um einen kryptografischen Angriff. Bei dieser Art von Angriffen wird das Ein-/Ausleseverfahren oder der Schlüssel angegriffen. Dafür gibt es diverse Möglichkeiten. Häufig wird jedoch der Brute-Force Angriff angewendet, bei welchem es sich um einfaches Ausprobieren aller möglichen Schlüssel handelt. Der Angriff auf den Auslesealgorithmus stellt eine weitere Möglichkeit dar. Bei diesem wird versucht herauszufinden, welche Bits mit den Daten markiert wurden und somit das Wasserzeichen repräsentieren. Durch das Wissen über die verwendeten Bits, können diese gezielt verändert werden, wodurch das Wasserzeichen zerstört wird. [37]

Ebenso häufig wird der Kollisionsangriff mehrerer Kunden angewendet. Dabei geht es darum, dass nicht nur der Urheber als Information eingefügt wird, sondern auch der Name des Kunden. Sobald dieser eine illegale Kopie des Dokuments erstellt und anschließend verteilt, kann zurück verfolgt werden, von wem das Dokument stammt und der Kunde somit zur Verantwortung gezogen werden. [33]

5 Vorstellung unterschiedlicher Untersuchungsmethoden

Es existieren zahlreiche Optionen, um Dateien miteinander zu vergleichen. Eine kurze Betrachtung der Textanalyse soll verdeutlichen, welche vielfältigen Ansätze existieren. Dennoch deckt dies nur einen kleinen Teil der gesamten Möglichkeiten ab. Einerseits kann ein Vergleich über die Inhaltsanalyse stattfinden. Dabei geht es vor allem darum, die Unterschiede innerhalb zwei verschiedener Dokumente festzustellen. Andererseits kann ein struktureller Vergleich stattfinden. Hierbei wird der Aufbau der Dokumente betrachtet. Es können Abschnitte, Überschriften oder Anordnungen von Tabellen untersucht werden. Das Ziel dabei ist es, Gemeinsamkeiten oder Unterschiede in der Strukturierung der Informationen festzustellen. [38, 39]

Diese Methoden sind für das Auffinden von digitalen Wasserzeichen nicht geeignet. Jedoch sinnvoll, um aufgefundene Wasserzeichen innerhalb der Dokumente einzuordnen und die Positionen zu bestimmen.

Eine andere Möglichkeit ist der Format- und Dateivergleich. Innerhalb dieses Verfahrens werden die Eigenschaften der Dateien betrachtet, wie beispielsweise das Dateiformat (PDF, Word-Dokument, Excel-Tabelle usw.), die Dateigröße, das Erstellungsdatum oder andere Metadaten.

Alternativ dazu gibt es die Variante, die Dokumente über mathematische Verfahren zu vergleichen. Hierzu zählt der Vergleich der Hashwerte sowie der Vergleich über die Byteebene. Bilddateien können zusätzlich über ihre Histogramme und ihre einzelnen Pixel verglichen werden.

Das Hauptaugenmerk der Arbeit liegt auf der Untersuchung von PDF-Dokumenten. Dabei ist die effektivste Methode, die Prüfung von Datenmengen über Metadaten oder mathematische Verfahren. Diese Verfahren können Hinweise auf Wasserzeichen geben und eignen sich damit besonders für das Auffinden von Identifikationsmerkmalen.

Auf Grundlage dessen geht es in diesem Kapitel um die drei Möglichkeiten Metadaten, Hashwerte und Byteebene. Diese sollen beschrieben und weiter diskutiert werden.

5.1 Metadaten

Die schnellste Methode, um zwei digitale Datenmengen (Dateien) miteinander zu vergleichen, ist der Vergleich über die Metadaten. Bei Metadaten handelt es sich um strukturierte Daten, die wichtige Informationen über Merkmale anderer Dateien geben. So beinhalten diese Hinweise zu Dokumenten, Bildern, Videos oder Internetseiten. Zum einfachen Verständnis kann ein Buch als Beispiel genommen werden, bei dem Autor, Titel und ISBN-Nummer als Metadaten hinterlegt sind, während der Inhalt des Buches als Datenmenge deklariert wird. Da

solche Daten maschinell auslesbar sind, können Crawler diese auslesen und Suchergebnisse dazu präsentieren. Ein Beispiel sind Systeme zur Literaturverwaltung. Mittels Metadaten zu einzelnen Büchern und Publikationen können diese einfach in Programme zur Literaturverwaltung eingegeben werden. Dies führt zu einer leichten Art und Weise, einzelne Werke ohne großen Aufwand zu finden. [40]

Metadaten können leicht über den Explorer auf einem Computer ausgelesen werden, denn es handelt sich dabei um die Eigenschaften, die zu einem Dokument, Bild oder Video gespeichert werden. Bei Dokumenten können beispielsweise Erstellungsdatum, Quelle und Anzahl der Seiten ausgelesen werden. Bei Bildern sind Informationen zu Bildgröße, Farbraum und teilweise auch zu Brennweite, Verschlusszeit oder Blende dargestellt (siehe Abbildung 5.1). Ebenso sind in den Metadaten die Rechte an dem jeweiligen Datenmaterial hinterlegt.

Abbildung 5.1: Beispiel Metadaten [eigene Darstellung]

Vorteile

Metadaten spielen eine wesentliche Rolle bei der Strukturierung und Organisation von Daten. Sie ermöglichen eine übersichtliche Darstellung und erleichtern das Auffinden von Informationen. Einer der Hauptvorteile der Verwendung von Metadaten besteht darin, dass unterschiedlichste Informationen gesammelt und in überschaubare Bereiche zusammengefasst werden können. Dies ermöglicht eine klare Strukturierung der Daten und erleichtert die Navigation durch große Datenmengen. Beispielsweise können Metadaten in Kategorien wie Datum, Autor, Dateityp, Größe oder Thema gruppiert werden. Dadurch wird es einfacher, die relevanten Daten zu identifizieren und auszuwählen. [40]

Ein weiterer wichtiger Aspekt von Metadaten ist die schnelle Zugriffszeit. Da die Metadaten zusätzliche Informationen über die Daten bereitstellen, können Anwendungen oder Systeme diese nutzen, um schnell auf die relevanten Daten zuzugreifen. Die Metadaten dienen als Referenzpunkte, welche die Suche beschleunigen und die Antwortzeiten verbessern können. Neben der schnellen Zugriffszeit ermöglichen Metadaten auch das Auslesen wichtiger Informationen zu den übergeordneten Dateien in kürzester Zeit. Beide Punkte sind besonders wichtig in Umgebungen, in denen große Datenmengen verarbeitet werden müssen und eine schnelle Datenabfrage oder Zusammenfassung von entscheidender Bedeutung ist, ohne die gesamten Dateien öffnen zu müssen. Durch den Zugriff auf die Metadaten können wichtige Eigenschaften oder Attribute der übergeordneten Dateien sofort abgerufen werden, was die Effizienz und den Arbeitsablauf erheblich verbessern kann. [40]

Zusammenfassend lässt sich sagen, dass Metadaten eine wertvolle Rolle bei der übersichtlichen Strukturierung von Daten spielen. Sie ermöglichen die Sammlung verschiedenster Informationen, ihre Kategorisierung und Indexierung sowie eine schnelle Zugriffszeit. Durch das Auslesen wichtiger Informationen zu den übergeordneten Dateien ermöglichen Metadaten eine effiziente Sichtung, Verwaltung und Organisation von Datenmengen.

Nachteile
Obwohl Metadaten viele Vorteile bieten, gibt es auch einige potenzielle Nachteile, die berücksichtigt werden müssen:

Datenschutz und Sicherheitsrisiken: Metadaten können sensible Informationen enthalten, die möglicherweise nicht für unbefugte Benutzer zugänglich sein sollten. Wenn Metadaten nicht angemessen geschützt werden, besteht die Gefahr von Datenschutz- oder Sicherheitsverletzungen. [41]

Datenqualität und -zuverlässigkeit: Metadaten sind abhängig von der Richtigkeit und Aktualität der zugrunde liegenden Daten. Wenn die Daten selbst unvollständig, inkonsistent oder fehlerhaft sind, kann dies zu falschen oder irreführenden Metadaten führen. Die Qualität und Zuverlässigkeit der Metadaten hängt also von der Qualität der zugrunde liegenden Daten ab. [41]

Aufwand für die Erstellung und Pflege: Metadaten müssen erstellt, aktualisiert und gepflegt werden, was zusätzlichen Aufwand erfordert. Insbesondere bei großen Datenmengen kann dies eine zeitaufwändige und ressourcenintensive Aufgabe sein. Metadaten müssen regelmäßig aktualisiert werden, sonst wird von veralteten Daten gesprochen und ihre Nützlichkeit ist nicht mehr gegeben. [42]

Komplexität und Standardisierung: Metadaten können in verschiedenen Formaten und Standards vorliegen, was zu Kompatibilitätsproblemen führen kann. Wenn verschiedene Systeme oder Anwendungen unterschiedliche Metadatenformate verwenden, kann dies die Interoperabilität und den Austausch von Daten erschweren. [43]

Begrenzter Kontext und Interpretation: Metadaten bieten zwar zusätzliche Informationen über die Daten, aber sie liefern keinen umfassenden Kontext. Einige Informationen können möglicherweise nicht in den Metadaten erfasst werden, was zu einer begrenzten Interpretation der Daten führen kann. Dies kann zu Missverständnissen oder falschen Schlussfolgerungen führen, wenn wichtige Informationen fehlen. [42]

Metadatenmanipulation: Dies bezeichnet die absichtliche Veränderung von Begleitinformationen zu Daten. Das wachsende Missbrauchspotenzial personenbezogener Daten resultiert aus der Fülle an verfügbaren Informationen und fortschreitenden Technologien zur Analyse. Metadaten, die oft als unscheinbare Begleitinformationen betrachtet werden, offenbaren tiefe Einblicke in das Leben von Personen. Diese können von Unbefugten für Identitätsdiebstahl und andere böswillige Zwecke genutzt werden. Da Metadaten unauffällig sind, lassen sie sich leicht manipulieren, um falsche Informationen zu schaffen. Es ist wichtig an-

zumerken, dass die Möglichkeit der Manipulation nicht nur den Dienstanbietern offensteht, welche die Datenhoheit haben. Tatsächlich können alle potenziell interessierten Einflussnehmer die Metadaten beeinflussen. [41]

Das Ziel der Arbeit liegt darin, PDF-Dokumente auf Wasserzeichen zu untersuchen. Metadaten geben die Grobstruktur an und sind für das Verständnis dafür eine Methode. Da es sich bei der Untersuchung jedoch um eine tiefergreifende Untersuchung handelt, sind Metadaten für das Auffinden von Wasserzeichen nicht geeignet. Sie können aber einen ersten Überblick über die Eigenschaften von Dateien geben.

5.2 Hashwerte

Eine weitere Option Dokumente zu vergleichen besteht darin, den Hashwert des Dokuments zu berechnen und mit einem bekannten Hashwert des originalen Dokuments oder des Wasserzeichens zu vergleichen.

Hashwerte sind ein kryptografisches Instrument, das in verschiedenen Bereichen der Informationstechnologie eingesetzt wird. Eine Hashfunktion, auch bekannt als Hash-Algorithmus, ist der Schlüssel zu diesem Prozess. Der Begriff „Hash“ lässt sich auf Deutsch mit „zerhacken“ übersetzen, was die Grundidee dahinter verdeutlicht. [44]

Die Hashfunktion nimmt eine beliebige Datenmenge in unterschiedlichen Formen entgegen, beispielsweise eine Textdatei, ein Bild oder eine andere Art von Datei. Diese Datenmenge wird von der Hashfunktion zerlegt und in eine neue Form gewandelt, die als Hashwert bezeichnet wird. Ein Hashwert ist eine unveränderbare Zeichenkette, die eine eindeutige Darstellung der ursprünglichen Daten oder Datenmenge darstellt. [44]

Für die Hashwerterstellung gibt es verschiedene Hashfunktionen, wie zum Beispiel MD5, SHA256 und SHA512. Diese nutzen verschiedene Vorgehensweisen und haben einen unterschiedlich langen Ausgabewert. Beispielsweise werden bei SHA256 256 Bits und bei SHA512 512 Bits verwendet. [44]

Eigenschaften und Vorteile
Ein wichtiger Aspekt von Hashwerten ist ihre einheitliche Länge. Unabhängig von der Größe oder Komplexität der Eingabedaten erzeugt eine Hashfunktion immer einen Hashwert mit derselben Länge. Dies ermöglicht es, den Hashwert als eine Art digitalen Fingerabdruck zu betrachten, der zur Identifizierung und Überprüfung der Integrität der Daten verwendet werden kann. [44]

Für die richtige Funktionsweise sind folgende Eigenschaften von erheblicher Bedeutung:

- Determinismus
- Geschwindigkeit
- Kollisionssicherheit
- Kontinuität bzw. Nichtkontinuität
- Nicht rücklesbar

Determinismus beschreibt die Länge des Hashwertes. Dieser muss bei unterschiedlichen Eingabelängen immer die gleiche Ausgabelänge besitzen. Zur Verdeutlichung sind in Tabelle 5.1 drei Syntagmen mit unterschiedlicher Länge bei der Eingabe (Klartext), aber gleicher Länge bei der Ausgabe (Hashwert) dargestellt. Weiterhin muss die gleiche Eingabe immer zur gleichen Ausgabe führen. [45]

Tabelle 5.1: Verdeutlichung Determinismus bei Hashwerten

Eingabetext	Hashwert SHA256
Heute scheint die Sonne und es wird sehr warm.	`93accd67dbf47987335d714a4b8f74491fb95f46004eb180a-d476ecb24b0a9df`
Das Wasser ist kalt.	`798684be6cf297c81a7c1ba1b6c7cde18ae94d59e8d1173e5-eb3e25a1305f134`
Ein Eis.	`6c6a19dc564fdd9f797d172abb60c2f9c7318344f99710073-593ff737f416f62`

Geschwindigkeit bezieht sich auf die Effizienz und Schnelligkeit, mit der ein kryptografischer Hash-Algorithmus in der Lage ist, den Hashwert einer gegebenen Eingabe zu berechnen. Ein schneller Hash-Algorithmus kann die Berechnung eines Hashwerts in kürzerer Zeit durchführen, während ein langsamer Algorithmus mehr Zeit benötigt. [45]

Die **Kollisionssicherheit** sagt aus, dass zwei unterschiedliche Eingaben nicht den gleichen Hashwert besitzen dürfen. Eine Kollision tritt auf, wenn zwei verschiedene Eingabewerte denselben Hashwert erzeugen. Jedoch können diese nicht effizient gefunden werden. Die Eingabemenge, welche beliebig groß sein kann, ist größer als die Ausgabemenge, welche begrenzt ist. Aus diesem Grund ist anzunehmen, dass mit hoher Wahrscheinlichkeit zwei Eingaben existieren, die über denselben Hashwert verfügen. Allerdings können diese beiden Werte nicht gefunden werden, da dafür Brute-Force notwendig ist und es diesbezüglich wiederum zu viele Möglichkeiten gibt. [45]

Bei Hashwerten bezieht sich **Kontinuität** auf die Eigenschaft, was eine Änderung in den Eingabedaten für Auswirkungen auf die Hashwerte haben. Kontinuität bedeutet, dass eine minimale Änderung in den Eingabedaten nur zu einer minimalen oder kaum wahrnehmbaren Änderung im Hashwert führt. **Nichtkontinuität** sagt aus, dass eine geringfügige Änderung in den Eingabedaten zu einer signifikanten Veränderung im generierten Hashwert führt. Mit anderen Worten, wenn auch nur ein einzelnes Zeichen der Eingabedaten geändert wird, ergibt sich ein völlig unterschiedlicher und nicht vorhersagbarer Hashwert (siehe Tabelle 5.2). Das heißt, wenn ein kleines Zeichen geändert wird, kann nicht vorher gesagt werden, wie sich der Hashwert ändert. [45]

Tabelle 5.2: Verdeutlichung Nichtkontinuität bei Hashwerten

Eingabetext	Hashwert SHA256
Heute scheint die Sonne und es wird sehr warm.	`93accd67dbf47987335d714a4b8f74491fb95f46004eb180-ad476ecb24b0a9df`
heute scheint die Sonne und es wird sehr warm.	`f408358e8a9f9ed4683b12340f51967bd3f6ef823308543b-ad054dcb840f3871`

Bei der letzten genannten Eigenschaft, **nicht rücklesbar**, geht es darum, dass aus einem berechneten Hashwert nicht der Originaltext abgeleitet werden kann (Einwegfunktion). [45] So kann in Tabelle 5.1 nicht von „6c6a19dc564fdd9f797d172abb60c2f9c7318344f99710073593ff737f416f62“ auf „Ein Eis.“ geschlossen werden.

Hashwerte helfen dabei, dass sich empfindliche Daten einfach und sicher speichern sowie verwalten lassen. Weiterhin bieten sie Sicherheit, da eine Rückführung nicht möglich ist und somit die Daten nicht einfach abgeleitet werden können. Wie bei der Geschwindigkeit schon genannt wurde, erfolgen Zugriffe schneller, was zu einer Zeitersparnis führt. Selbst wenn Hashwerte abgefangen oder geklaut werden, haben die Angreifer keinen Nutzen davon, da sie mit diesen keine Rückschlüsse auf Informationen bekommen können. Zuletzt helfen Hashwerte dabei, dass die Integrität von Daten und Nachrichten sicher und zuverlässig geprüft werden kann. [45]

Nachteile

Die unterschiedlichen Eigenschaften einer Hashfunktion können jedoch schnell zu Nachteilen werden. So sorgt die deterministische Eigenschaft dafür, dass für dieselben Eingabedaten immer derselbe Hashwert erzeugt wird. Dies bedeutet, dass bei Verwendung von Hashwerten in bestimmten Anwendungen wie Passwort-Speicherung, identische Passwörter denselben Hashwert erzeugen, was ein Sicherheitsrisiko darstellen kann. Dort sichert die Eigenschaft nicht rücklesbar zu sein, dass von einem Hashwert nicht auf die ursprünglichen Eingabedaten geschlossen werden kann. Dies kann in einigen Fällen jedoch problematisch sein, wenn die Originaldaten später benötigt werden. Weiterhin handelt es sich bei einer Hashfunktion nicht um eine Verschlüsselung. Hashwerte sind unidirektional und werden häufig für die Gewährleistung der Integrität und nicht für die Vertraulichkeit von Daten verwendet. Auch die Kollision kann als Nachteil betrachtet werden, weil sie die Integrität und Zuverlässigkeit von kryptografischen Hashfunktionen beeinträchtigt. So wird es einem Angreifer ermöglicht, absichtlich eine bösartige Nachricht zu erstellen, welche denselben Hashwert wie eine legitime Nachricht hat. Dies führt dazu, dass die Integrität von Daten kompromittiert wird, da es schwierig wird, zwischen den legitimen und gefälschten Inhalten zu unterscheiden. Zuletzt sollte noch die teilweise hohe Rechenleistung erwähnt werden. Bei der Verwendung von starken kryptografischen Hashfunktionen kann die Berechnung der Hashwerte rechenintensiv sein, insbesondere bei großen Datenmengen. Dies kann die Leistung in einigen Anwendungsfällen beeinträchtigen. [45]

Neben Eigenschaften und Vorteilen ist die Anwendung ein ebenso entscheidender Faktor. So haben Hashwerte zahlreiche Anwendungen in der Informationssicherheit. Folgend wird lediglich auf den für die Arbeit benötigten Schwerpunkt eingegangen. Sie werden häufig verwendet, um sicherzustellen, dass die Daten während der Übertragung oder Speicherung nicht verändert wurden. Dies wird über den Vergleich des Hashwertes realisiert. [45]

Zusammenfassend kann gesagt werden, dass Hashwerte insgesamt ein unverzichtbares Instrument in der Kryptografie sind und eine effiziente Möglichkeit bieten, Daten in kompakter Form zu repräsentieren und deren Integrität zu überprüfen. Es ist eine sehr schnelle und sichere Methode und der Angriff ist durch seine Einwegfunktion fast unmöglich.

Eine Auswertung der Hashwerte ist über Tabellenform möglich, das heißt, dass eine schnelle Übersicht und Kontrolle, ob die Eingaben identisch sind, gegeben ist. Vorhandene Wasserzeichen können somit über eine Änderung innerhalb des Hashwertes erkannt werden.

Anwendung

Um die Hashwerte der einzelnen Dateien zu berechnen, wurde ein Programm in Python[1] erstellt. Dabei handelt es sich um eine weit verbreitete und universelle Programmiersprache. Sie wird auch als Skriptsprache bezeichnet, da ein erstelltes Programm interpretiert wird, wenn es gestartet wird. [46]

Der Programmcode ist im folgenden Quelltext 5.1 dargestellt und wird im Anschluss genauer erklärt. Grundsätzlich soll er dazu dienen, den SHA256-Hash von zwei PDF-Dateien zu berechnen und anschließend zu vergleichen. Die Ergebnisse sollen in einer Textdatei festgehalten werden.

```
1  import hashlib, sys, os
2  from datetime import datetime
3
4  def compute sha256 hash(file path):
5      sha256 hash = hashlib.sha256()
6      with open(file path, "rb") as f:
7          while chunk := f.read(8192):
8              sha256 hash.update(chunk)
9      return sha256 hash.hexdigest()
10
11 def main():
12     if len(sys.argv) != 3:
13         print("Nutzung: python compare files.py <path to file1> <path to file2
       >")
14         return
15
16     file1 path = os.path.abspath(sys.argv[1])
17     file2 path = os.path.abspath(sys.argv[2])
18
19     file1 name = os.path.basename(file1 path)
20     file2 name = os.path.basename(file2 path)
21
22     if not os.path.exists(file1 path):
23         print(f"Error: Datei '{file1 path}' existiert nicht.")
24         return
25     if not os.path.exists(file2 path):
26         print(f"Error: Datei '{file2 path}' existiert nicht.")
27         return
28
29     try:
30         hash1 = compute sha256 hash(file1 path)
31         hash2 = compute sha256 hash(file2 path)
32
33         result folder = "results"
```

[1] https://www.python.org

```
34          if not os.path.exists(result_folder):
35              os.makedirs(result_folder)
36
37          timestamp = datetime.now().strftime("%Y-%m-%d %H-%M-%S")
38          result_filename = f"{result_folder}/{file1_name} {file2_name} {
        timestamp}.txt"
39
40          comparison_result = hash1 == hash2
41
42          with open(result_filename, "w") as f:
43              f.write(f"Hash von {file1_name}: {hash1}\n")
44              f.write(f"Hash von {file2_name}: {hash2}\n")
45              f.write(f"Ergebnis: {'Gleich' if comparison_result else 'Nicht
        gleich'}\n")
46
47      except Exception as e:
48          print(f"Ein Fehler ist aufgetreten: {e}")
49
50  if __name__ == "__main__":
51      main()
```

Quelltext 5.1: Berechnung und Vergleich des Hashwertes

In Zeile 1 und 2 werden die benötigten Module importiert. Die Funktion `compute_sha256_hash()`, welche in Zeile 4 beginnt, berechnet den SHA256-Hash einer Datei. Zusätzlich liest sie die Inhalte in Chunks, so genannte Teilinhalte, ein, damit auch große Dateien verarbeitet werden können. Am Ende gibt sie den Hexadezimalwert des Hashes zurück.

Ab Zeile 11 beginnt der Hauptteil des Skripts, die `main()`-Funktion. Zuerst überprüft sie, ob das Skript mit den richtigen Befehlszeilenargumenten aufgerufen wurde (zwei Dateipfade werden erwartet). Zeile 16 und 17 dienen dazu, dass relative und absolute Pfade verarbeitet werden können. Anschließend werden in Zeile 19 und 20 die Dateinamen aus dem Pfad extrahiert.

Ab der Zeile 22 werden verschiedene Überprüfungen und Aktionen durchgeführt:

- Zeile 22 bis 27: Es wird überprüft, ob beide angegebenen Dateien existieren.
- Zeile 29 bis 31: Die SHA256-Hashes beider Dateien werden berechnet.
- Zeile 33 bis 35: Ein Ordner namens „results“ wird erstellt, wenn er nicht existiert.
- Zeile 37: Ein Zeitstempel wird erstellt.
- Zeile 38: Ein Dateiname für die Ergebnisdatei wird generiert, der den Namen der beiden Dateien sowie den Zeitstempel enthält.
- Zeile 40 bis 45: Die Hashes und das Ergebnis der Dateivergleichsoperation werden in die Ergebnisdatei geschrieben.

Sofern etwas bei der Hashberechnung nicht funktioniert haben sollte, wird in Zeile 47 eine Fehlermeldung ausgegeben. Schließlich wird die `main()`-Funktion aufgerufen, wenn das Skript direkt ausgeführt wird.

Zusammengefasst führt dieses Programm die folgenden Schritte aus:

1. Dateipfade und Dateinamen werden überprüft.
2. Die SHA-256-Hashes der Dateien werden berechnet.
3. Ein Ordner für die Ergebnisdateien wird erstellt.
4. Ein Zeitstempel wird erstellt.
5. Die Ergebnisse des Hashvergleichs werden in eine Ergebnisdatei geschrieben.

Durch den Vergleich der SHA256-Hashwerte soll festgestellt werden, ob die Dateien identisch sind oder durch das Herunterladen von zwei verschiedenen Benutzerkonten Änderungen am Datenmaterial durchgeführt wurden.

5.3 Byteebene

Eine weitere Möglichkeit, Dateien zu vergleichen, ist der Vergleich auf Byteebene. Die Byteebene ist eine noch tiefere Ebene der Datenverarbeitung. Ein Byte ist die kleinste darstellbare Einheit von Informationen in einem Computer. Die Analyse auf Byteebene kann aufschlussreiche Informationen über Dateien, Strukturen und potenzielle Fehler liefern.

Dabei wird der Inhalt zweier Dateien miteinander verglichen und Unterschiede können festgehalten und ausgegeben werden. Innerhalb der Ausgabe sind die vorhandenen Unterschiede dargestellt und es wird angegeben, was geändert werden müsste, damit beide Dateien übereinstimmen.

Es gibt eine Vielzahl an Möglichkeiten, Dateien auf Byteebene zu vergleichen. Im Folgenden sind zwei genannt, auf welche im Anschluss näher eingegangen wird:

1. mittels Linux Programm *diff*
2. mittels eigenem Programm

Das diff-Tool kann verwendet werden, um Unterschiede zwischen Text- oder binären Dateien zu finden. Es kann jedoch manchmal schwierig sein, die gefundenen Unterschiede in binären Dateien zu interpretieren. Aus diesem Grund gibt es spezialisierte Tools, die dafür geeigneter sind, wie zum Beispiel Visual Binary Diff (VBinDiff).

5.3.1 Linux Programm *diff*

Linux bietet ein Standardprogramm an, welches ein Teil der GNU Diffutils und somit in jeder Installation vorhanden ist. Das Programm ist unter dem Namen *diff* bekannt. Damit können Inhalte von zwei Dateien verglichen werden oder auch Ordner, um herauszufinden, ob sie gleiche Dateien enthalten. Mit dem Aufruf des Programms werden zwei Dateien eingelesen, welche anschließend verglichen werden sollen. Bei dem Vergleich beider Dateien wird ausgegeben, an welchen Stellen sich diese unterscheiden und was geändert werden muss, damit sie identisch sind. Es wird also angegeben, was in der einen Datei (Datei1) geändert werden muss, um die andere Datei (Datei2) zu erhalten. Dabei können Zeilen abgeändert, gelöscht oder hinzugefügt werden. [47]

Die Syntax des Programms lautet: `diff [Optionen] datei1 datei2`.

Das Optionenfeld ist fakultativ verwendbar und bietet verschiedene Möglichkeiten bei der Ausgabe. In der folgenden Tabelle 5.3 sind einige diese Optionen gelistet und beschrieben.

Tabelle 5.3: Aufruf Optionen des Programms diff [47]

Optionen	Beschreibung
-q	quick: meldet, wenn Dateien unterschiedlich sind, unterdrückt aber die Ausgabe der Unterschiede
-s	same: meldet, wenn Dateien gleich sind `diff -qs` beschränkt die Ausgabe auf die reine Meldung
-r	rekursiv: vergleicht Unterverzeichnisse, wenn vorhanden
-y	tabellarische Ausgabe, gleiche und ungleiche Zeilen werden markiert
-a	behandelt alle Dateien (zum Beispiel binäre) wie Text
-d	versucht mit erhöhtem Aufwand kleinere Veränderungen zu finden

Nach der Veranschaulichung der Möglichkeiten sind folgend zwei Beispiele zum Verständnis dargeboten.

Der Inhalt von zwei Dateien ist in Abbildung 5.2 dargestellt. Diese sollen in den Beispielen miteinander verglichen werden, um mögliche Unterschiede aufzudecken.

```
1 Zeile eins
2 Zeile zwei
3 Zeile drei
4 Zeile fuenf
5 Zeile sieben
```

```
1 Zeile eins
2 Zeile drei
3 Zeile vier
4 Zeile fuenf
5 Zeile sechs
```

Abbildung 5.2: Inhalt von Datei1 und Datei2

Beispiel 1: `diff Datei1 Datei2`

Der Aufruf bewirkt, dass zwei Dateien (Datei1 und Datei2) miteinander verglichen werden. Mögliche Unterschiede werden auf der Konsole ausgegeben. Diese Ausgabe ist in Abbildung 5.3 dargestellt. Aus dieser lässt sich ableiten, was in Datei1 geändert werden muss, damit sie mit Datei2 übereinstimmt. Die Ausgabe wird folgend noch näher erklärt.

Zeilen, die am Anfang „**<**“ stehen haben sind nur in der ersten Datei und Zeilen mit „**>**“ am Anfang sind nur in der zweiten Datei vorhanden.

Als erstes muss die zweite Zeile von Datei1 gelöscht (d - delete) werden, damit beide Dateien ab Zeile eins synchron sind. Im Anschluss wird in die dritte Zeile von Datei1 die dritte Zeile von Datei2 hinzugefügt (a - add) und abschließend noch die fünfte Zeile von Datei1 in die fünfte Zeile von Datei2 umgeändert (c - change).

```
parallels@debian-gnu-linux-11:~$ diff Datei1 Datei2
2d1
< Zeile zwei
3a3
> Zeile vier
5c5
< Zeile sieben
---
> Zeile sechs
```

Abbildung 5.3: Beispiel für Aufruf `diff Datei1 Datei2`

Beispiel 2: `diff -q Datei1 Datei2`

Wie in Tabelle 5.3 aufgelistet ist, bewirkt die Option -q im Aufruf, dass die Ausgabe von möglichen Unterschieden unterdrückt wird und nur ausgegeben wird, ob die Dateien unterschiedlich sind oder nicht. Dies ist in Abbildung 5.4 abgebildet.

```
parallels@debian-gnu-linux-11:~$ diff -q Datei1 Datei2
Files Datei1 and Datei2 differ
```

Abbildung 5.4: Beispiel für Aufruf `diff -q Datei1 Datei2`

5.3.2 Eigenes Programm

Neben der Möglichkeit, zwei Dateien mittels des Linux Programms *diff* zu vergleichen, gibt es ebenfalls die Möglichkeit eines selbst erstellten Programms. Dieses Programm wurde ebenfalls in Python geschrieben und ist in Quelltext 5.2 abgebildet.

```python
def compare_files(file1, file2):
    with open(file1, 'rb') as f1, open(file2, 'rb') as f2:
        while True:
            byte1 = f1.read(1)
            byte2 = f2.read(1)
            if byte1 != byte2:
                return False
            if not byte1:
                # Both files reached the end simultaneously
                return True

datei1 = "<path to file>"
datei2 = "<path to file>"

result = compare_files(datei1, datei2)
if result:
    print("Die Dateien sind byteweise identisch.")
else:
    print("Die Dateien weisen Unterschiede auf.")
```

Quelltext 5.2: Vergleich von Dateien auf Byteebene

Der oben genannte Python-Programmcode dient dazu, zwei Dateien auf Byteebene zu vergleichen, um festzustellen, ob sie identisch sind oder nicht. Dabei wird jedoch nur festgestellt, ob es Unterschiede gibt und nicht, wo sich diese befinden bzw. welche Unterschiede es sind.

In Zeile 1 befindet sich eine Funktion mit dem Namen `compare_files` und den Aufrufparametern *file1* und *file2*. Die Funktion wird definiert, um den Vergleich durchzuführen. Dabei werden zwei Dateipfade (file1 und file2) als Eingabe erwartet.

In Zeile 2 werden die Dateien eingelesen. Dafür dient der Aufruf `open(textdatei.txt, 'MODUS')`.

Bei MODUS gibt es erneut verschiedene Möglichkeiten, die verwendet werden können. Grundsätzlich bestimmt der Modus, welche Aktionen mit der eingelesenen Datei durchgeführt werden dürfen. Folgend sind die möglichen Optionen mit Beschreibung gelistet:

Tabelle 5.4: Möglichkeiten für Dateien beim Einlesen [46]

Option	Beschreibung
w	write: nur schreiben (bestehender Inhalt wird überschrieben)
a	append: wird an bestehenden Inhalt angehängt
r	read: nur für Lesen
r+	Lesen und Schreiben
b	in Binärform für Lesen und Schreiben (die anderen Modi werden durch b ergänzt)

Im Code wird in Zeile 2 folgender Aufruf benutzt: *with open(file1, 'rb') as f1, open(file2, 'rb') as f2*. Damit werden beide Dateien in einem kontextbasierten Manager geöffnet, um sie im binären Modus zu lesen (rb steht für read binary). Ebenfalls wird Datei1 in der Variable f1 und Datei2 in der Variable f2 gespeichert.

In der Zeile 3 bis 10 verwendet der Code eine Endlosschleife (*while True*), um die Dateien byteweise zu vergleichen. In jedem Schleifendurchlauf liest der Code ein Byte von beiden Dateien mit *byte1 = f1.read(1) und byte2 = f2.read(1)*. Die beiden gelesenen Bytes (byte1 und byte2) werden in Zeile 6 mit *if byte1 != byte2* verglichen. Wenn sie nicht übereinstimmen, wird *False* zurückgegeben, was bedeutet, dass die Dateien nicht identisch sind. Wenn beide Bytes gleich sind, wird in Zeile 8 mittels *not byte1* geprüft, ob das Ende der Dateien erreicht ist. Trifft das zu, haben beide Dateien gleichzeitig das Ende erreicht und es wird *True* zurückgegeben. Das wiederum bedeutet, dass die Dateien identisch sind. Falls weder eine Abweichung noch das Ende der Dateien erreicht wurde, geht der Code zur nächsten Iteration der Schleife über und vergleicht die nächsten Bytes. Der Code wird so lange ausgeführt, bis entweder eine Abweichung festgestellt wird oder beide Dateien das Ende erreichen. Nachdem die Schleife beendet ist, wird die Funktion beendet und der Rückgabewert *True* oder *False* zurückgegeben, in Abhängigkeit, ob die Dateien identisch sind oder nicht.

Anschließend werden in Zeile 12 und 13 zwei Variablen erstellt, in denen jeweils die zu untersuchende Datei übergeben wird.

Zum Schluss folgt in Zeile 15 der Aufruf der Funktion mit den übergebenen Dateien. Das Ergebnis davon wird in die Variable *result* geschrieben. Als Ausgabe wurden Aussagen definiert, ob die Dateien byteweise identisch sind oder nicht.

Es muss beachtet werden, dass der Code in binärer Form arbeitet, um Dateien byteweise zu vergleichen. Daher ist er für den Vergleich von Textdateien oder Dateien mit bestimmten Codierungen nicht geeignet. Er eignet sich besser, um originale binäre Dateien, wie zum Beispiel Bilder oder Audios zu vergleichen.

Abschließend kann für die Untersuchung auf Byteebene gesagt werden, dass diese mehr Ressourcen benötigt als die Untersuchung der Hashwerte. Der Nachteil dieser Methode ist, dass das Betriebssystem Linux benötigt wird, um sie auszuführen. Sie ist dennoch hilfreich, um Unterschiede, die bereits bekannt sind, näher zu betrachten.

5.4 Diskussion

Metadaten, Hashwerte und die Byteebene haben eine zentrale Bedeutung in der digitalen Welt, indem sie Datenverwaltung, Sicherheit und Integrität ermöglichen. Jedes dieser Konzepte hat einzigartige Funktionen und Vorteile, aber auch potenzielle Herausforderungen, die es beim Vergleichen von Dateien zu berücksichtigen gilt.

Nach der Vorstellung der einzelnen Optionen mit ihren Vor- und Nachteilen sowie deren Eigenschaften soll in diesem Unterkapitel festgelegt werden, welche davon für eine Untersuchung angewendet werden können und sollen. Dafür spielen die Eigenschaften eine entscheidende Rolle, aber auch die Einsatzgebiete und das angestrebte Ziel.

Metadaten sind Informationen, die Daten begleiten und ihnen Kontext verleihen. Sie können beispielsweise den Autor, das Erstellungsdatum, den Dateityp und vieles mehr enthalten. Sie sind wertvoll für die effiziente Organisation und Kategorisierung von Daten in Datenbanken und Archiven. Sie ermöglichen auch eine verbesserte Suchfunktionalität, da Suchmaschinen und Datenbanken Metadaten nutzen, um relevante Ergebnisse zu liefern.

Bei der Verwendung von Metadaten ist es leicht, diese anzusehen und auszuwerten. Da jedoch in unterschiedlichen Dateiarten verschiedene Metadaten gespeichert werden, kann nicht pauschal immer mit den gleichen Inhalten verglichen werden. Folglich ist hier zu beachten, dass nur Dateien der gleichen Ursprungsart verglichen werden können. Vorteilhaft ist die schnelle Zugriffszeit und, dass von einem Computer mittels Datei-Explorer einfach darauf zugegriffen werden kann. Abhängig vom Internetbrowser kann es passieren, dass die Metadaten unvollständig und fehlerhaft sind. Dies spricht gegen eine Verwendung dieser Möglichkeit zur Prüfung auf Wasserzeichen. Die Gefahr falscher Ergebnisse ist durch Vorliegen von fehlerhaften Grundvoraussetzungen sehr hoch. Es gibt keinen Standard in dem Metadaten vorliegen müssen und welche Informationen diese enthalten. Weiterhin sind Metadaten nicht immer vorhanden und abrufbar. Ebenso kann das Risiko bestehen, durch die Arbeit mit veralteten Metadaten zu falschen Ergebnissen zu kommen. Auch die Manipulation von Metadaten ist möglich, sodass aus diesem Grund ebenfalls mit falschen Ergebnissen analysiert werden könnte.

Metadaten sind wichtig zum Auffinden von Internetseiten, aber gleichzeitig risikobehaftet, da sie auch sensible Informationen über den Benutzer preisgeben können, zum Beispiel Standort oder Kontaktdaten. Dies spielt jedoch für die Betrachtung von PDF-Dokumenten keine Rolle. Daher ist es eine Option, die Metadaten zu analysieren, um einen ersten Überblick über die Dateien und deren Eigenschaften zu erhalten.

Anders als bei Metadaten lässt sich der Hashwert für jede Datei berechnen, ohne das bestimmte Dateien oder Informationen vorhanden sein müssen. Hashwerte sind eine Art digitaler Fingerabdruck für Daten. Sie werden durch kryptografische Hashfunktionen erzeugt und sind eindeutig für jeden Datensatz.

Vorteile der Hashwerte liegen bei der jeweils definierten einheitlichen Länge und den entsprechenden Eigenschaften. Jedoch muss in dem Zusammenhang ebenfalls erwähnt werden, dass einige der Eigenschaften sowohl Vor- als auch Nachteile bilden können.

Einerseits ist es von Vorteil und wichtig, dass von den Hashwerten nicht auf die Daten geschlossen werden kann. Andererseits kann es nachteilig sein, wenn die Eingabedaten nicht mehr verfügbar sind und somit nicht mehr mit den Originaldaten gearbeitet werden kann. Die Geschwindigkeit der Erstellung spricht jedoch für die Verwendung von Hashwerten, da damit schnell und sicher zwei Dateien miteinander verglichen werden können.

Durch die Sicherheit, dass ein kleiner Unterschied in den Daten zu einem völlig anderen Hashwert führt, ist die Möglichkeit, ein Wasserzeichen aufzudecken, sehr hoch. Hashwerte sind nützlich, um die Integrität von Daten sicherzustellen und ihre Unveränderlichkeit zu überprüfen. Dies ermöglicht es, sicherzustellen, dass Daten während der Übertragung oder Speicherung nicht manipuliert wurden und im Umkehrschluss Manipulationen damit aufgedeckt werden können. Diese Eigenschaft wird zukünftig bei Signatur- und Authentifizierungsverfahren eine immer größere Rolle erhalten. Bereits jetzt werden Hashwerte häufig verwendet, um Passwörter zu speichern, ohne die tatsächlichen Passwörter preiszugeben.

Damit ist Methode des Hashwertvergleichs gut geeignet, um Wasserzeichen innerhalb von PDF-Dokumenten aufzudecken.

Neben Hashwerten können Dateien auf ihrer Byteebene untersucht werden. Allerdings ist die Analyse auf Byteebene oft technisch anspruchsvoll und erfordert spezialisierte Kenntnisse beziehungsweise entsprechende Programme.

Innerhalb der hier vorgenommenen Analyse wird jeweils ein Byte mit einem anderen Byte verglichen und untersucht, ob es sich dabei um den gleichen Inhalt handelt.

Insgesamt sind Metadaten, Hashwerte und die Byteebene unverzichtbare Werkzeuge für die Verwaltung, Sicherheit und Analyse von Daten in der digitalen Welt. Aus diesem Grund sollen die folgenden Techniken bei der kommenden Analyse in Kapitel 6 angewendet werden.

6 Untersuchung von PDF-Dokumenten auf digitale Wasserzeichen

Zuerst wird ein grober Überblick mittels der Metadaten verschafft. Dabei liegt das Augenmerk darauf, womit der Inhalt erstellt wurde, und ob bzw. welche Codierungs-Software verwendet wurde. Da nicht alle Dateien die gleichen Metadaten besitzen, geht es bei dieser Gegenüberstellung lediglich um einen ersten Einblick über die Dateien.

Nach den Metadaten soll ein Vergleich über die Hashwerte der Dateien stattfinden. Diese werden zuerst für jedes Dokument berechnet, um die Werte anschließend zu vergleichen. Dafür soll das erstellte Programm aus Abschnitt 5.2 (siehe Quelltext 5.1) verwendet werden. Da der Hashwert für jedes Dokument einmalig ist und somit nicht zwei verschiedene Dokumente den gleichen Hashwert besitzen können, werden die Ergebnisse für eine bessere Auswertbarkeit direkt in eine Datei geschrieben. Treten bei dem Vergleich der Hashwerte Unterschiede auf, ist das ein Indiz dafür, dass digitale Wasserzeichen vorhanden sind. Die Dateien, welche unterschiedliche Hashwerte aufweisen, werden anschließend auf Byteebene untersucht. Damit soll analysiert werden, worin sich die Dateien unterscheiden und ob es sich dabei tatsächlich um ein digitales Wasserzeichen handeln könnte.

Die zuvor beschriebenen Vorgehensweisen zum Dokumentenvergleich werden in diesem Kapitel vorbereitet und folgend auf unterschiedliche Dateien angewendet und die Vergleichsergebnisse ausgewertet.

6.1 Vorbereitung

Bevor mit einer Analyse begonnen werden kann, werden der Rahmen und die Bedingungen dafür festgelegt. Im Rahmen der vorliegenden Arbeit soll überprüft werden, ob digitale Wasserzeichen in digitalen Dokumenten immer vorhanden sind, von bestimmten Faktoren abhängen oder in den untersuchten Anwendungsgebieten noch gar nicht eingesetzt werden. Es sollen die folgenden verschiedenen **Arten** von Dokumenten untersucht werden:

- Bücher
- Zeitschriften
- Artikel
- Hochschulschriften
- Paper

Um die verschiedenen Dokumente zu erhalten, wurde das Rechercheportal der Hochschulbibliothek Mittweida (PRIMO)[2] genutzt.

In den folgenden Abschnitten werden die heruntergeladenen Objekte als **Dokumente** bezeichnet. Die einzelnen Downloads der Dokumente über die verschiedenen Benutzerkonten werden als **Dateien** benannt. So wird beispielsweise das Objekt mit dem Namen „Fusing

[2] `https://mit-primo.hosted.exlibrisgroup.com/primo-explore/search?vid=MIT_VU1`

Blockchain and AI With Metaverse A Survey“ als Dokument bezeichnet und die einzelnen Downloads dieses Dokuments als Dateien. Zum besseren Verständnis dient das Schema in Abbildung 6.1.

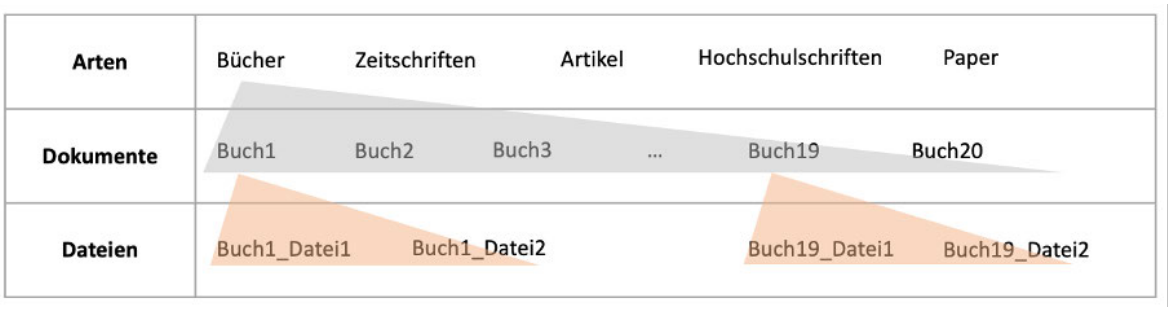

Abbildung 6.1: Bildliche Darstellung der eingeführten Begriffe: Arten, Dokumente, Dateien [eigene Darstellung]

Innerhalb einer Art wurden 20 verschiedene Dokumente von zwei unterschiedlichen Benutzerkonten heruntergeladen, sodass insgesamt 40 Dateien einer Kategorie vorhanden sind, welche jeweils verglichen werden können. Aus Zeitgründen und der begrenzten Verfügbarkeit des zweiten Benutzerkontos wurde die Anzahl der untersuchten Dokumente einer Art auf 20 begrenzt. Die Objekte wurden von zwei verschiedenen Internetanbietern heruntergeladen. Datei1 wurde mittels des Browsers „Safari“und Datei2 mittels des Browsers „Mozilla Firefox“ heruntergeladen.

Mit Kenntnis der Struktur eines PDF-Dokumentes (siehe Abschnitt 2.1), kann vermutet werden, dass eingebettete Wasserzeichen in dem Bereich des Bodys zu finden sind. Das Augenmerk der Untersuchung sollte auf diesem Bereich liegen.

6.2 Durchführung

Alle Dokumente sollen auf die gleiche Art und Weise untersucht werden. Die Metadaten werden dabei lediglich für einen ersten Überblick herangezogen, da diese für die untersuchten Medien jeweils gleich sein sollten. Dass Hinweise auf Wasserzeichen mittels der Metadatenanalyse aufgefunden werden, ist unwahrscheinlich. Anschließend sollen die Hashwerte für die Dateien berechnet und miteinander verglichen werden. Falls dabei Unterschiede festgestellt werden, sollen die Dokumente auf Byteebene untersucht werden, um herauszufinden wo sich Unterschiede befinden und welche das sind.

6.2.1 Bücher

Bevor mit der Durchführung der Bücher begonnen werden kann, wird festgelegt, welche Bücher untersucht werden sollen. Hierfür wurden zwei Oberkategorien bestimmt. Zum einen sollen die Bücher eines konkreten Verlags untersucht werden und ob dieser die Bücher mit Wasserzeichen kennzeichnet. Zum anderen sollen Bücher nach ihrem Erscheinungsjahr analysiert werden, um die Frage zu klären, ob neue Bücher bereits ein Wasserzeichen besitzen.

Bei der Untersuchung von Büchern nach einem Verlag wurde sich für den **Springer-Verlag** entschieden. Die Bücher, welche dabei heruntergeladen wurden, stehen in keinem thematischen Zusammenhang, sondern wurden alle zufällig gewählt. Dies trifft auch auf die Untersuchung von Büchern nach einem bestimmten Erscheinungsjahr zu. Dabei wurde das **Jahr 2022** als Erscheinungsjahr für die zu untersuchenden Bücher festgelegt. Diese Bücher stammen wiederum von unterschiedlichen Verlagen, sodass die beiden Kategorien nicht ineinander übergreifen. Eine Auflistung der untersuchten Bücher ist im Abschnitt A.1 zu finden.

Bei beiden Kategorien werden zuerst die Metadaten mittels Datei-Explorer analysiert, ob dort bereits Unterschiede zwischen den PDF-Dateien erkennbar sind. Als Datei-Explorer wird „Finder“ in Version 13.2 verwendet. Im Anschluss daran wird der Hashwert mittels des Programms in Quelltext 5.1 für alle Dateien berechnet, verglichen und zur besseren Übersicht in eine Textdatei geschrieben. Die anschließende Auswertung ist in Unterabschnitt 6.3.1 dargestellt.

6.2.2 Zeitschriften

Die analysierten Dokumente lassen sich der Kategorie der Zeitschriften zuordnen. Alle Dateien wurden mittels des Suchbegriffs *Computer* auf dem Rechercheportal der Hochschule aufgefunden und das Erscheinungsjahr wurde auf die letzten fünf Jahre begrenzt. Die Auswahl erfolgte willkürlich und steht in keinem weiteren Zusammenhang. Im Abschnitt A.2 ist eine Übersicht über die untersuchten Zeitschriften abgebildet.

Bei diesen Dokumenten wurden ebenfalls zuerst die Metadaten mittels „Finder“ als Datei-Explorer analysiert. Im Anschluss daran wurden die Hashwerte mittels des Programms in Quelltext 5.1 der einzelnen Dateien berechnet und miteinander verglichen. Bei Unterschieden wurden die Dokumente zusätzlich auf Byteebene untersucht, um herauszufinden, wo die Unterschiede liegen. Dafür kam das Linux Programm *diff* zur Anwendung, da dieses Informationen liefert, was in einem Dokument geändert werden muss, damit zwei Dokumente übereinstimmen. Da das Programm die Unterschiede lediglich auf der Konsole ausgibt, wurden sie für eine bessere Auswertung in eine Textdatei geschrieben. Die Auswertung der Unterschiede erfolgt ausführlich im Unterabschnitt 6.3.2.

6.2.3 Artikel

Bei Artikeln handelt es sich um allgemeine schriftliche Arbeiten, die in verschiedenen Kontexten verwendet werden können. Bei der Auswahl der Dokumente wurde ebenfalls der Suchbegriff *Computer* verwendet sowie das Erscheinungsjahr auf die letzten fünf Jahre beschränkt. Die Auswahl der Dokumente erfolgte willkürlich und steht in keinem Zusammenhang. Eine Übersicht über die untersuchte Literatur ist in Abschnitt A.3 zu finden.

Als erster Schritt wurden wieder die Metadaten mittels „Finder“ untersucht. Im Anschluss wurden mittels des Programms in Quelltext 5.1 die Hashwerte berechnet und miteinander verglichen. Dabei wurden die Ergebnisse in eine Textdatei geschrieben, um die Auswertung

zu vereinfachen. Anschließend erfolgte auch hier der Vergleich über die Byteebene unter Verwendung des Linux Programms *diff*. Die Ergebnisse wurden in separaten Textdateien erfasst. Die Auswertung dieser wird in Unterabschnitt 6.3.3 genauer beschrieben und dargestellt.

6.2.4 Hochschulschriften

Bei dieser Kategorie handelt es sich um Arbeiten, welche im Rahmen von Abschlussprojekten an der Hochschule Mittweida erstellt und veröffentlicht wurden. Die heruntergeladenen Dokumente sind Arbeiten, deren Erscheinungsjahre innerhalb der letzten fünf Jahre liegen. Diese sollten dahingehend untersucht werden, ob bereits Wasserzeichen enthalten sind. Auch hierbei wurde als Suchbegriff *Computer* angewendet. Eine Übersicht der untersuchten Arbeiten ist im Abschnitt A.4 zu finden.

Nach dem Download der Dokumente wurden mittels des Datei-Explorers „Finder“ die Metadaten analysiert. Im Anschluss daran wurden mit dem Programm, welches im Quelltext 5.1 abgebildet ist, die Hashwerte berechnet und verglichen. Das Ergebnis ist in Unterabschnitt 6.3.4 dargestellt.

6.2.5 Paper

Paper sind speziellere schriftliche Arbeiten, die hauptsächlich in der akademischen Welt verwendet werden. Dabei sollten Forschungsergebnisse und wissenschaftliche Erkenntnisse präsentiert werden. Der Suchbegriff *Computer* wurde hier ebenfalls festgelegt und die Erscheinungsjahre der Dokumente liegen zwischen 2018 und 2023. Die Auswahl der Dokumente erfolgte zufällig. Diese sind im Abschnitt A.5 abgebildet.

Analog zu den bisherigen Untersuchungen erfolgt auch hier die entsprechende Vorgehensweise. Zuerst wurden die Metadaten mittels „Finder“ analysiert. Anschließend erfolgte die Berechnung und der Vergleich der Hashwerte mit dem Programm, welches im Quelltext 5.1 dargestellt ist. Im Unterabschnitt 6.3.5 ist das Ergebnis der Untersuchung beschrieben.

6.3 Auswertung

In diesem Unterkapitel werden die Ergebnisse der zuvor durchgeführten Untersuchungen vorgestellt. Dabei sollen Aussagen zu Metadaten, Hashwerten und der Byteebene getroffen werden. Diese sind verbal formuliert und als Tabelle im Anhang B zu finden.

Innerhalb der Auswertung liegt der Fokus auf der Untersuchung von vorhandenen Wasserzeichen. Die zuvor gemachten Analysen helfen dabei, diese zu finden und auszuwerten. Das Programm, für die Berechnung und den Vergleich der Hashwerte, hat die jeweiligen Ergebnisse in die gleiche Datei geschrieben. Aus diesem Grund ist die Auswertung übersichtlich zusammengefasst und einheitlich durchzuführen.

6.3.1 Bücher

Aufgrund der Untersuchung von zwei Kategorien finden auch zwei voneinander getrennte Auswertungen statt. Einerseits die Untersuchung auf Verlag-Ebene und andererseits die Untersuchung in Abhängigkeit des Jahres.

Springer-Verlag
Da nach eingehender Analyse keine Unterschiede innerhalb der Metadaten vorhanden waren, führte deren Sichtung zu keinem Ergebnis.

In Quelltext B.1 ist zu sehen, dass die Dateien pro Dokument den gleichen Hashwert aufweisen. Das bedeutet, dass innerhalb der Dateien keine Unterschiede vorhanden sind, egal von welchem Browser oder mit welchem Benutzerkonto die Dateien heruntergeladen wurden. Auf Grundlage dessen kann davon ausgegangen werden, dass sich kein digitales Wasserzeichen in den Dokumenten befindet, welches mit den Benutzerkonten oder Browsern zusammenhängt. Es besteht dennoch die Möglichkeit, dass ein Wasserzeichen hinterlegt ist, welches die Urheberschaft belegt.

Jahr 2022
Auch in dieser Kategorie führte das Untersuchen der Metadaten zu keinem Ergebnis. Ebenfalls wiesen die Dateien zu einem Dokument jeweils die gleichen Hashwerte auf. Diese sind in Quelltext B.2 abgebildet. Aufgrund der jeweiligen Hashwerte pro Dokument wird gezeigt, dass Browser und Benutzer keine Relevanz aufweisen. Auch hier besteht die Möglichkeit, dass ein Wasserzeichen hinterlegt ist, welches den Urheber bestätigt. In der Untersuchung wurde kein Wasserzeichen gefunden, das auf den Benutzer hinweist, der die Datei heruntergeladen hat.

6.3.2 Zeitschriften

Bei der Untersuchung von Zeitschriften wurde festgestellt, dass bei einigen Dateien, welche mit dem Browser „Mozilla Firefox“ heruntergeladen wurden, eine Quelle in den Metadaten hinterlegt wurde. Beispielhaft ist dies in Abbildung 6.2 abgebildet. Bei der Rückprüfung dieser hinterlegten Quelle wurde kein Treffer gefunden und der Zugriff war nicht möglich.

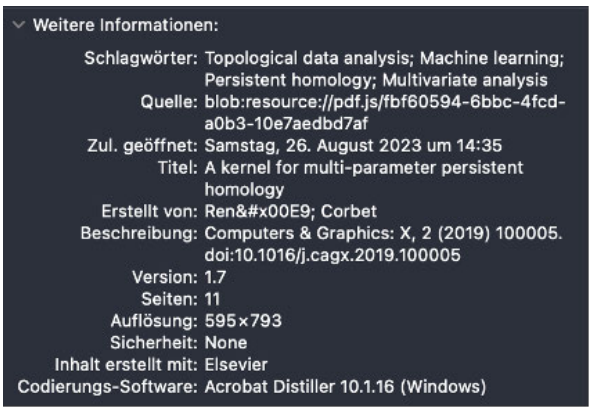

Abbildung 6.2: Beispiel: Metadaten, in denen eine Quelle hinterlegt ist [eigene Abbildung]

Nachdem die einzelnen Hashwerte berechnet wurden, waren Unterschiede schnell und einfach zu erkennen. Die Hashwerte sind in Abschnitt B.2 dargestellt. Dateien, die unter einer Zahlenfolge (zum Beispiel 1-s2.0-S2590148619300056-main_Datei1.pdf) gespeichert wurden, wiesen alle unterschiedliche Hashwerte auf.

Im Gegensatz dazu zeigten Dateien, die unter ihrem Titel gespeichert wurden, gleiche Hashwerte, mit Ausnahme von einer Datei. Aufgrund dieser Auffälligkeit innerhalb der Hashwerte wurde die nachfolgende Untersuchung auf Byteebene durchgeführt.

Die Ergebnisse von der Untersuchung der Byteebene werden nachfolgend erläutert. Zuerst wird die Auswertung der einzelnen Dokumente aufgelistet. Im Anschluss folgt eine Übersicht über gefundene Gemeinsamkeiten und was diese aussagen.

Zeitschrift1
Innerhalb von Zeitschrift1 waren verschiedene Hashwerte vorhanden. Die Untersuchung auf Byteebene ergab, dass sich beide Dateien lediglich in einer Zeile unterscheiden. Durch das Linux Programm *diff* konnte der Unterschied in Zeile 3.678 festgestellt werden. Wie Abbildung 6.3 zeigt, müsste die Zeile 3.678 in Datei1 in die Zeile 3.678 aus Datei2 umgeändert werden. Durch die Zeilen, die jeweils nur in Datei1 bzw. Datei2 vorhanden sind, kann geschlossen werden, dass durch das Herunterladen von verschiedenen Benutzerkonten eine Änderung am Datenmaterial vollzogen wurde. Dies gibt Hinweise auf ein Wasserzeichen, welches durch den Download von einem Benutzerkonto eingebettet wurde.

```
3678c3678
<           <LMfnsotuPyPr.ywmGy.uNzcNNnPn-lwiQy9yGmd2NmtiPytmRzgf-o9ePotaNnd6OnM2Tma/>
---
>           <Ml8ZFnMeLyM2Rnd6GnMn9nsNNntaKlt6Jod-GyPj-y92Rn.uRyM-No9ePotaNnd6OmMyTma/>
```

Abbildung 6.3: Unterschied innerhalb von Zeitschrift1 [eigene Abbildung]

Zeitschrift2
Zeitschrift2 weist die gleichen Unterschiede wie Zeitschrift1 auf. Nach der Untersuchung auf Byteebene wurde festgestellt, dass sich die Dateien bei diesem Dokument ebenfalls nur in einer Zeile unterscheiden. Um die gleichen Dateien zu erhalten, müsste Zeile 6.021 in Datei1 mit der gleichen Zeile aus Datei2 abgeändert werden. Durch diesen Unterschied kann ebenfalls vermutet werden, dass innerhalb des Datenmaterials Informationen, in Form eines Wasserzeichens, zu den unterschiedlichen Benutzerkonten hinterlegt wurden. Weiterhin wurde festgestellt, dass die Zeile aus Datei1 die gleichen ersten fünf Buchstaben wie Datei1 von Zeitschrift1 besitzt.

Zeitschrift3
Dieses Dokument weist Unterschiede in der Zeile 10.536 auf. Auch dabei müsste diese Zeile von Datei1 in die Zeile von Datei2 geändert werden, damit die Dateien übereinstimmen. Der Rest der Dateien stimmt überein. Folglich scheint in dieser Zeile ein Hinweis auf das Benutzerkonto hinterlegt zu sein. Der Hinweis auf das Benutzerkonto ist in Form eines Wasserzeichens in das Datenmaterial eingebettet.

Zeitschrift4
Bei diesem Dokument wurde ein Unterschied in der Zeile 10.272 aufgefunden. Auch hier muss diese Zeile lediglich umgeändert werden, um die andere Datei zu erhalten. Dies kann ebenfalls ein Hinweis auf ein Wasserzeichen sein, in dem das Benutzerkonto innerhalb der Datei hinterlegt wurde.

Zeitschrift5
Innerhalb der Untersuchung auf Byteebene wurden mehrere Unterschiede festgestellt. Zum einen ist bei diesem Dokument ebenfalls wieder eine Zeile (8.305) vorhanden, die lediglich abgeändert werden müsste, um die andere Datei zu erhalten. Auch hierbei wird es sich um ein Wasserzeichen handeln, welches Informationen im Datenmaterial eingebettet hat.

Zusätzlich zu der Zeile gibt es einen weiteren Unterschied. Dieser zeigt an, dass innerhalb von Datei1 die Zeilen von 9.244 bis 9.429 gelöscht werden müssten, um Datei2 zu erhalten. Damit kann davon ausgegangen werden, dass Datei1 eine längere Datei als Datei2 ist.

Zeitschrift6
Bei diesem Dokument wurde als Ergebnis der Untersuchung auf Byteebene nur ein Unterschied in der Länge der Dateien festgestellt. So verfügt Datei1 über 51 Zeilen mehr als Datei2. Das Linux Programm *diff* gibt hier aus, dass diese Zeilen gelöscht werden müssten, damit Datei1 identisch mit Datei2 ist.

Gemeinsamkeiten der untersuchten Zeitschriften
Nachdem in Abschnitt 2.1 der Aufbau eines PDF-Dokumentes vorgestellt wurde, sollen festgestellte Gemeinsamkeiten im Bezug darauf dargelegt werden.

Die Dokumente, welche unter einer Zahlenfolge gespeichert wurden (Zeitschrift1 bis Zeitschrift5), stammen vom gleichen Herausgeber und wurden von der gleichen Seite heruntergeladen. Der Herausgeber der Zeitschriften ist Elsevier[3]. Durch diese Gemeinsamkeit kann ein erster Zusammenhang festgestellt werden. Drei der fünf Dokumente stammen aus dem Jahr 2019, die beiden anderen aus dem Jahr 2020. Dies zeigt auf, dass seit mindestens 2019 Änderungen am Datenmaterial beim Herunterladen vorgenommen werden.

Durch die Betrachtung, wie ein PDF-Dokument aufgebaut ist, konnten die unterschiedlichen Zeilen untersucht werden, was zu der Feststellung führte, dass sich die Zeilen immer im gleichen Objekt befinden. Dieses Objekt ist jeweils das letzte im Body der Dateien.

Dazu fiel auf, dass die Zeilen von allen Zeitschriften, die jeweils geändert werden müssten, Gemeinsamkeiten aufwiesen. So verfügen alle über eine identische Zeichenfolge von elf Zeichen, die sich kurz vor Zeilenende befindet. Im Anschluss sind vier beliebige Zeichen und es folgen nochmal vier Zeichen, die wieder bei allen identisch sind. Durch diese Gemeinsamkeiten besteht die Möglichkeit, dass es sich bei einer der Zeichenfolgen um die Hinterlegung der Hochschule handelt und bei der anderen um die Lizenz, welche die Hochschule für diese Seite besitzt.

[3] `https://www.elsevier.com/de-de`

Bei der genauen Betrachtung von Zeitschrift5 und Zeitschrift6 fiel auf, dass bei Datei1 beider Zeitschriften der gleiche, folgende Aufbau festgestellt werden konnte: *Header, Body, Referenztabelle, Trailer, Body, Referenztabelle, Trailer*. Dies würde bedeuten, dass nach dem ersten PDF-Dokument noch ein weiteres folgt, jedoch ohne Header.

Um einen möglichen Grund dafür zu finden, wurde Datei1 von Zeitschrift5 und Zeitschrift6 nochmal mit dem Browser „Mozilla Firefox“ heruntergeladen. Bei Zeitschrift5 wiesen beide Dateien noch einen unterschiedlichen Hashwert, jedoch die gleiche Länge auf. Zusätzlich dazu gab es eine Änderung innerhalb der Zeile 8.305. Dabei stimmten die zuvor übereinstimmenden elf Zeichen nur noch zum Teil überein. Dies kann ein Hinweis auf ein dynamisches Wasserzeichen sein. Ein dynamisches Wasserzeichen verändert sich bei jeder Kopie des Dokuments. Dies kann mithilfe von Pseudozufallszahlengeneratoren erreicht werden. Aufgrundlage dessen kann vermutetet werden, dass dieser Teil der Zeichenfolge einen Hinweis auf die Anzahl der Downloads oder den Zeitpunkt des Downloads gibt.

Bei Zeitschrift6 war das Ergebnis ähnlich. Nach der Berechnung des Hashwertes, wiesen beide Dateien den gleichen Hashwert auf, was ebenfalls zu einer gleichen Länge führt, das heißt es handelt sich jetzt um identische Dateien. Aus dieser Erkenntnis kann geschlussfolgert werden, dass der Browser „Safari“ Änderungen am Datenmaterial vollzogen hat und nicht das Benutzerkonto.

6.3.3 Artikel

Bei der Untersuchung der Metadaten fiel auf, dass die Dateien, welche mittels des Browsers „Mozilla Firefox“ heruntergeladen wurden, eine Quelle hinterlegt hatten. Diese war jedoch im Anschluss nicht mehr aufrufbar.

Bei dem im nächsten Schritt erfolgten Vergleich über die Hashwerte wurden ebenfalls Unterschiede sichtbar. In Abschnitt B.3 ist eine Übersicht über die Hashwerte zu finden. Bei Dateien, deren Dateiname aus einer Zahlenfolge besteht, fanden sich häufiger Unterschiede, als bei Dateien, die unter ihrem Titel gespeichert wurden. Aufgrund der Unterschiede innerhalb der Dateien folgte die weitere Untersuchung auf Byteebene. Die Ergebnisse dieser sind folgend für jeden Artikel separat gelistet. Im Anschluss findet auch hier eine Übersicht gefundener Gemeinsamkeiten und ihren Bedeutungen statt.

Artikel1
Die Untersuchung von Artikel1 ergab einen Unterschied in der Zeile 2.290. Hier müsste lediglich eine Zeile in die andere umgeändert werden, damit beide Dateien identisch sind. Aus dem Grund kann bei diesem Unterschied auf ein Wasserzeichen geschlossen werden, in dem Hinweise auf die verschiedenen Benutzerkonten vorhanden sind.

Artikel2
Der Unterschied innerhalb von Artikel2 wurde in Zeile 1.720 festgestellt. Da verschiedene Zeilen in beiden Dateien vorhanden sind, ist dies ein Indiz dafür, dass in den Zeilen jeweils die Benutzerkonten hinterlegt sind, mit welchen die Dateien heruntergeladen wurden. Die Zeilen können als hinterlegtes Wasserzeichen betrachtet werden.

Artikel3
Hierbei wurde ein Unterschied in der Zeile 1.306 deutlich. Dies lässt Rückschlüsse darauf ziehen, dass eine Änderung am Datenmaterial durch die Benutzerkonten erfolgte, da mit dem Umschreiben einer Zeile beide Dateien übereinstimmen würden. Die Veränderungen, welche durch die Benutzerkonten vorgenommen wurden, deuten auf ein vorhandenes Wasserzeichen hin.

Artikel4
Auch hier geben die Unterschiede innerhalb der Zeilen Hinweise darauf, dass durch die Benutzerkonten Änderungen vorgenommen wurden. Der Unterschied wurde in Zeile 2.281 gefunden und kann ein Indiz auf ein hinterlegtes Wasserzeichen sein. Der Beginn der Zeile in Datei2 stimmt mit dem Anfang der Zeile von Datei2 in Artikel1 überein.

Artikel5
Bei der Untersuchung kam ein Unterschied in Zeile 3.202 zum Vorschein, in der beide Dateien verschieden sind. Dies ist eine Andeutung dafür, dass Hinweise auf beide Benutzerkonten in Form eines Wasserzeichens innerhalb der Zeilen hinterlegt sind.

Artikel6
Der Unterschied wurde in der Zeile 11.733 festgestellt. Der Anfang der Zeile in Datei1 entspricht dem Beginn der Zeilen in Artikel1 und Artikel5 von Datei1. Aufgrund dessen kann auf einen Zusammenhang zwischen den Zeilen und somit auf Änderungen durch ein Benutzerkonto geschlossen werden. Diese Änderungen wurden in Form eines Wasserzeichens innerhalb des Datenmaterials hinterlegt.

Artikel7
Das Dokument weist Unterschiede in der Zeile 1.865 auf. Auch hierbei muss die eine Zeile in einer Datei lediglich abgeändert werden, damit sie mit der anderen Datei übereinstimmt. Daraus können Rückschlüsse auf ein Wasserzeichen gezogen werden, welches durch die Benutzerkonten erstellt wurde.

Artikel8
Es befindet sich eine Änderung in der Zeile 1.639, die dazu führt, dass die Dateien nicht identisch sind. Sofern eine von diesen umgeändert wird, stimmen beide Dateien überein. Deshalb kann davon ausgegangen werden, dass beim Herunterladen durch verschiedene Benutzerkonten ein Vermerk hinterlegt wurde, mit welchem Konto der Download stattfand. Dieser Vermerk kann als Wasserzeichen betrachtet werden.

Es fand sich eine Gemeinsamkeit innerhalb der abzuändernden Zeile von Datei2 der Artikel Artikel8 und Artikel2 sowie der Datei1 des Artikels Artikel4. Diese Gemeinsamkeit umfasst die ersten fünf Buchstaben und lässt sich über die Hinterlegung von Benutzerdaten nicht erklären.

Bei den bisherigen Artikeln Artikel1 bis Artikel8 ist ebenso aufgefallen, was bereits bei den Unterschieden innerhalb der Zeitschriften festgestellt wurde. Die letzten vier Zeichen der jeweiligen abzuändernden Zeilen in den Dateien stimmen bei allen Artikeln überein, was ein

Indiz auf die Hochschule Mittweida sein kann. Weiterhin ist aufgefallen, dass in den Zeilen eine Zeichensequenz vorhanden ist, die in jeder Datei auftritt und sich kurz vor dem Ende befindet, dabei kann es sich um eine Lizenz für den Herausgeber handeln.

Artikel9
Die Unterschiede, die innerhalb der Untersuchung von Artikel9 aufgefallen sind, weichen von den bisherigen ab. Innerhalb der Unterschiede gibt es nicht eine Zeile, die umgeändert werden müsste. Es sind eine Vielzahl von unterschiedlichen, teils kleinen Änderungen über das komplette Datenmaterial vorhanden. Die vorhandenen Unterschiede geben einen Hinweis darauf, dass durch die Benutzerkonten Änderungen vorgenommen wurden. Jedoch sind diese vereinzelt über die gesamte Byteebene der Datei zu finden, wodurch eine zentrale Aussage über die Änderung erschwert wird. Bei den jeweiligen Unterschieden handelt es sich augenscheinlich nur um Zahlen.

Artikel10
Bei der Untersuchung von Artikel10 sind ebenfalls mehrere Unterschiede zum Vorschein gekommen. Teils handelt es sich dabei um kleine Änderungen innerhalb einer Zeile und teils um Änderungen, die über mehrere Zeilen hinweg durchgeführt werden müssen.

Artikel11
Der Artikel weist innerhalb der Untersuchung ähnliche Unterschiede auf, wie Artikel9 und Artikel10. Die Unterschiede, die vorhanden sind, erstrecken sich über mehrere Zeilen. Bei einem erneuten intensiven Vergleich wurde festgestellt, dass die Positionen der Objekte in der Referenztabelle jeweils um zwei Bytes verschoben sind.

Gemeinsamkeiten der untersuchten Artikel
Wie bereits erwähnt wurde, weisen alle Artikel, deren Dateiname eine Zahlenfolge ist, unterschiedliche Hashwerte auf. Auffällig ist, dass alle betroffenen Artikel von der gleichen Internetseite bezogen und vom selben Herausgeber veröffentlicht wurden. Dabei handelt es sich um den gleichen Herausgeber wie bei den Zeitschriften. Gleichzeitig wurde festgestellt, dass die Artikel ebenfalls im Jahr 2023 publiziert wurden. Über diese Gemeinsamkeiten lassen sich erste Zusammenhänge feststellen.

Bei allen Artikeln gab es Unterschiede, welche auf die einzelnen Benutzerkonten zurückzuführen sind. Analog zu den Zeitschriften befinden sich auch hier die Unterschiede, bis auf wenige Ausnahmen, an den gleichen Stellen.

Unter Bezugnahme auf den Aufbau einer PDF-Datei (siehe Abschnitt 2.1) befinden sich die jeweils unterschiedlichen Zeilen von Artikel1 bis Artikel8 im vorletzten Objekt des Bodys. Die Anzahl der Objekte innerhalb eines Dokumentes variiert, was dazu führt, dass die Zeilenzahl und die Objektnummer jeweils eine andere ist.

Bei den Artikeln Artikel9 bis Artikel11 fanden sich Unterschiede teilweise an mehreren Stellen und in verschiedenen Ausprägungen. Besonders auffallend war hier, dass sich die Unterschiede über den gesamten Body zeigten. Es scheint als würden ganze Objekte fehlen oder vertauscht seien.

Ebenfalls fiel hier auf, dass die Zeilen von allen Artikeln, die jeweils geändert werden müssten, Gemeinsamkeiten aufweisen. So verfügen alle über eine Zeichenfolge von elf Zeichen, die sich kurz vor Zeilenende befindet. Im Anschluss sind vier beliebige Zeichen und es folgen vier Zeichen, die wieder bei allen identisch sind. In diesem Vergleich fiel auch auf, dass von den bereits erwähnten elf Zeichen die ersten sieben bei Zeitschriften und Artikeln gleich sind. Auch die letzten vier Zeichen sind bei beiden Dokumentarten identisch. Das lässt Rückschlüsse daraus ziehen, dass es sich bei den Zeichenfolgen um die Hinterlegung der Hochschule und um den Herausgeber handelt.

Durch die unterschiedlichen Hashwerte der Artikel Artikel1 bis Artikel8 ist die Vermutung naheliegend, dass es sich dabei um Wasserzeichen handeln könnte. Eine genauere Untersuchung dieser ist in Unterabschnitt 6.3.6 dargestellt.

6.3.4 Hochschulschriften

Bei der Sichtung der Metadaten der Hochschulschriften fanden sich keine Unterschiede. Auch bei der Analyse der Hashwerte, inklusive der Vergleiche, konnten keine Unterschiede erkannt werden. Die berechneten Hashwerte sind in Abschnitt B.4 dargestellt. Aus diesem Grund lässt sich schlussfolgern, dass bei den untersuchten Dokumenten (Abschlussarbeiten) von der Hochschule Mittweida keine digitalen Wasserzeichen hinterlegt sind.

6.3.5 Paper

In den Metadaten der Paper gab es keine Hinweise auf Unterschiede und somit auch keine Hinweise auf mögliche Wasserzeichen. Auch die Berechnung der Hashwerte wies identische Werte auf. Dies kann in Abschnitt B.5 nachvollzogen werden. Das erlaubt die Schlussfolgerung, dass innerhalb der untersuchten Paper keine Wasserzeichen eingebettet sind.

6.3.6 Untersuchung der aufgefundenen Wasserzeichen

Bei den aufgefundenen Wasserzeichen handelt es sich hauptsächlich um eine Zeile innerhalb der Dateien. Aus diesem Grund sind diese auf verschiedene Weisen angegriffen und getestet worden, um mehr darüber zu erfahren.

Der erste Angriff war der Löschangriff. Dabei wurden die PDF-Dateien zuerst mittels einem Texteditor geöffnet. Anschließend wurden die abzuändernden Zeilen der jeweiligen Datei gesucht und entfernt. Dies sollte einen Rückschluss auf den Urheber und das Benutzerkonto ausschließen. Die Zeilen ließen sich einfach, schnell und ohne weiteren Aufwand entfernen. Nach der Entfernung dieser Zeilen, ließen sich die Dokumente dennoch normal öffnen und die Anzeige erfolgte ohne weitere Probleme. Weiterhin wiesen sie nach der Entfernung identische Hashwerte auf.

Genauso verhielt es sich bei der Doppelmarkierung. Bei diesem Angriff wurde das Wasserzeichen von einer Datei in eine andere Datei zusätzlich eingefügt. Beispielsweise wurde das Wasserzeichen der Datei2 von Zeitschrift1 in Datei1 von Zeitschrift1 eingefügt. Das Ergebnis war hier analog zum Löschangriff. Die Dateien ließen sich normal und ohne Probleme öffnen und betrachten.

In beiden Fällen waren keine sichtbaren Veränderungen an der PDF-Datei zu erkennen. Auch das normale Wasserzeichen ist nicht sichtbar in den Dateien.

Auf Grundlage, dass sich die hinterlegten Zeilen einfach entfernen lassen, ohne sichtbare Änderungen vorzunehmen, kann darauf geschlossen werden, dass es sich bei den Wasserzeichen nicht um robuste Wasserzeichen handelt. Jedoch lässt sich die Eigenschaft der Nicht-Wahrnehmbarkeit feststellen, da die hinterlegten Zeilen innerhalb der PDF-Dateien mit durchschnittlichen Sehvermögen nicht erkannt werden können. Zusätzlich scheint es sich um einen relativ einfachen Algorithmus zu handeln, da das Wasserzeichen schnell erkannt werden konnte und leicht durch Angriffe zu entfernen war.

Bei der Betrachtung der Objektgröße, in denen das Wasserzeichen eingefügt ist, konnte festgestellt werden, dass die Objekte bei Zeitschriften circa 5.900 Bytes und bei Artikeln circa 6.500 Bytes umfassen. Auffallend bei den Artikeln war, dass im Objekt kurz vor Ende viele Leerzeichen eingefügt waren, was eine Erklärung für die unterschiedlichen Größen darstellt. Der Nutzen dieser Leerzeichen konnte nicht festgestellt werden. Im Vergleich dazu umfasst das eingefügte Wasserzeichen in allen Dokumenten lediglich 72 Bytes.

Durch die Untersuchung der Wasserzeichen ist zu vermuten, dass es sich dabei um dynamische Wasserzeichen handelt. Bei einem weiteren Download der gleichen Datei über das gleiche Benutzerkonto war das enthaltene Wasserzeichen ein anderes. Deswegen liegt die Vermutung nah, dass darin der Zeitpunkt oder die Anzahl der Downloads hinterlegt sind.

Testweise erfolgte die Einbettung eines der gefundenen Wasserzeichen in ein beliebiges PDF-Dokument, bei dem dadurch keinerlei Veränderungen feststellbar waren.

7 Fazit und Ausblick

Der Bereich der digitalen Wasserzeichen ist ein relativ junger Bereich der Informatik. Bis vor wenigen Jahren war die Nutzung vorwiegend auf Bild- und Tonmaterial zur Absicherung von Urheberrechten beschränkt. Inzwischen findet es Anwendung in allen Bereichen, wie in Dokumenten oder Videos.

Das Ziel dieser Arbeit war es, digitale Literatur bezüglich unsichtbarer digitaler Wasserzeichen zu untersuchen. Dabei sollte geklärt werden, in welcher Art von Dokumenten diese bereits vorhanden sind und was darin gespeichert ist. Der Fokus lag dabei auf den Unterschieden zwischen zwei Dateien, welche mittels zwei verschiedener Benutzerkonten bezogen wurden, und ob Rückschlüsse auf die Konten gezogen werden können.

Die umfassende Analyse hat gezeigt, dass innerhalb von Büchern, Hochschulschriften und Papern keine digitalen Wasserzeichen vorhanden sind, von denen auf verschiedene Benutzerkonten geschlossen werden kann. Es ist jedoch möglich, dass sich in diesen Arten von Dokumenten Wasserzeichen befinden, welche Informationen über den Urheber beinhalten. Eine Analyse dieser Informationen geht über den Umfang dieser Arbeit hinaus.

Im Vergleich dazu sind bei Zeitschriften und Artikeln deutliche Unterschiede erkennbar gewesen. Nach der Berechnung der Hashwerte der Dokumente, welche Unterschiede aufzeigten, wurde die Untersuchung auf Byteebene durchgeführt. Diese zeigt die Gründe für die unterschiedlichen Hashwerte auf. Nach intensiver Analyse auf dieser Ebene kann gesagt werden, dass innerhalb der Dokumente Wasserzeichen vorhanden sind. Bei diesen Wasserzeichen handelt es sich um dynamische Wasserzeichen, da sich diese bei weiteren Downloads verändert haben. Wichtig ist, dass einige Änderungen nicht durch das Benutzerkonto verursacht wurden, sondern durch den Browser, über welchen die Dateien heruntergeladen wurden.

Die hinterlegten Wasserzeichen ließen sich jedoch schnell und einfach entfernen, sodass nach einer Entfernung keine Rückschlüsse mehr möglich sind. Das spricht gegen die Verwendung solcher Wasserzeichen, die diesen Einbettungsalgorithmus verwenden. Denn dieser hinterlegt die Informationen lediglich als eine Zeile innerhalb des Bodys der Dateien und werden bei Änderung des Dateiformates mit bloßem Auge sichtbar.

Zusammengefasst kann gesagt werden, dass die Nutzung von unsichtbaren digitalen Wasserzeichen bereits erfolgt. Innerhalb der hier betrachteten PDF-Dokumente sind die Eigenschaften der Wasserzeichen jedoch informationstechnisch schwach umgesetzt. Offensichtlich besteht Notwendigkeit und Potenzial, um Manipulationen von Dokumenten zu verhindern und das Gleichgewicht zwischen Schutz und Lesbarkeit zu erhalten.

Zukünftig kann auf die bisherigen Ergebnisse aufgebaut und mit einem größeren Querschnitt der Objekte (mehr Dokumente und mehr Kategorien) weitere Erkenntnisse gewonnen werden. Wichtig ist dabei auch die Auswahl weiterer und anderer Untersuchungsmethoden.

Durch die technische Entwicklung gewinnt die Absicherung geistigen Eigentums gegen Diebstahl und Fälschung sowie der Schutz und die nachweisbare Kennzeichnung von Urheberrechten zunehmend an Bedeutung. Wahrscheinlich wird dieser Bereich auch für die Forensik immer bedeutender, da die Anzahl und die Schwere der Straftaten voraussichtlich zunehmen werden. Ähnlich dazu wird mutmaßlich der Nachweis von digitalen Wasserzeichen für Urheberrechte, Fälschungen und Diebstahl in Strafverfahren zukünftig eine wichtige Rolle spielen und immer mehr an Bedeutung gewinnen.

Vorstellbar ist, dass die Rolle digitaler Wasserzeichen als Identifikationsmerkmal über alle Bereiche stetig wachsen und durch die zunehmende Digitalisierung des Alltags eine breitere Anwendung der Wasserzeichen unabdingbar sein wird. Mit der umfassenden Anwendung besteht eine größere Gefahr von Angriffen und deren Abwandlungen. Daher muss die Technologie der digitalen Wasserzeichen immer weiterentwickelt werden, um deren Eigenschaften zu gewährleisten.

Neue Ansätze, die auf kryptografischen Methoden oder maschinellem Lernen basieren, fordern die Wirksamkeit und Vielseitigkeit dieser Identifikationsmerkmale zu steigern. Zukünftige Forschungen müssen sich daher auf innovative Ansätze zur Verbesserung der Wasserzeichen-Technologie konzentrieren.

Anhang A: Untersuchte Literatur

A.1 Bücher

```
978 1 4020 5604 8.pdf:
  Titel: The Sun and Space Weather
  Autor: Arnold Hanslmeier

978 1 4614 4924 9.pdf:
  Titel: Cobalt Blues
  Autor: Peter R. Almond

978 1 4614 5632 2.pdf:
  Titel: 3D Surface Reconstruction
  Autor: Francesco Bellocchio, N. Alberto Borghese, Stefano Ferrari, Vincenzo Piuri

978 3 642 00150 5.pdf:
  Titel: Signaling Pathways in Liver Diseases
  Autor: Jean Francois Dufour, Pierre Alain Clavien

978 3 030 36271 3.pdf:
  Titel: Children's Exploration and Cultural Formation
  Autor: Mariane Hedegaard

978 3 030 37177 7.pdf:
  Titel: Data Journeys in the Sciences
  Autor: Sabina Leonelli

978 3 030 41694 2.pdf:
  Titel: Nostalgia and Hope: Intersections between Politics of Culture, Welfare, and Migration in Europe
  Autor: Ov Cristian Norocel, Anders Hellström

978 3 030 45559 0.pdf:
  Titel: The Prevent Duty in Education
  Autor: Joel Busher, Lee Jerome

978 3 030 49679 1.pdf:
  Titel: Beyond Media Borders, Volume 1
  Autor: Lars Elleström

978 3 030 51237 8.pdf:
  Titel: Migration and Social Protection in Europe and Beyond (Volume 3)
  Autor: Jean Michel Lafleur, Daniela Vintila

978 3 030 54871 1.pdf:
  Titel: Civilian Lunatic Asylums During the First World War
  Autor: Claire Hilton

978 3 030 66303 2.pdf:
  Titel: Parenting and Work in Poland
  Autor: Katarzyna Suwada

978 3 319 57846 0.pdf:
  Titel: Grid Generation Methods
  Autor: Vladimir D. Liseikin

978 3 642 02608 9.pdf:
  Titel: Optimality and Risk Modern Trends in Mathematical Finance
  Autor: Freddy Delbaen, Christophe Stricker

978 3 642 17005 8.pdf:
  Titel: Vector Optimization
  Autor: Johannes Jahn

978 3 662 08542 4.pdf:
  Titel: Mathematical Biology
  Autor: J.D.Murray

978 3 658 31994 6.pdf:
  Titel: Soziale Arbeit und Sucht
  Autor: Marcel Krebs, Roger Mäder, Tanya Mezzera

978 981 15 0364 1.pdf:
  Titel: Workers, Managers, Productivity
  Autor: Akio Hosono, John Page, Go Shimada

978 981 15 6540 3.pdf:
  Titel: China: Surpassing the "Middle Income Trap"
  Autor: Shaojie Zhou, Angang Hu

978 981 15 7865 6.pdf:
  Titel: Global History with Chinese Characteristics
  Autor: Manuel Perez Garcia
```

Quelltext A.1: Metadaten der Bücher vom Springer-Verlag

```
978 981 19 3763 7.pdf:
  Titel: Analysis of Reaction Diffusion Models with the Taxis Mechanism
  Autor: Yuanyuan Ke, Jing Li, Yifu Wang

978 3 662 64435 5.pdf:
  Titel: Fließgewässer  und Auenentwicklung
  Autor: Heinz Patt

978 3 658 36403 8.pdf:
  Titel: Arbeitsrecht
  Autor: Tim Jesgarzewski

978 3 658 37005 3.pdf:
  Titel: Vergaberecht
  Autor: Daniel Naumann

978 3 658 38163 9.pdf:
  Titel: Öffentliche Finanzen und Verhaltensökonomik
  Autor: Thomas Döring

978 3 658 35995 9.pdf:
  Titel: Pflegeeinrichtungen erfolgreich führen
  Autor: Heinrich Bolz

978 3 658 34970 7.pdf:
  Titel: Aktives Altern im digitalen Zeitalter
  Autor: Susanne Ring Dimitriou, Minas Dimitriou

978 3 658 28253 0.pdf:
  Titel: Strukturanalyse der Gegenwart
  Autor: René König

978 3 030 96454 2.pdf:
  Titel: Landscapes of Lifelong Learning Policies across Europe
  Autor: Sebastiano Benasso, Dejana Bouillet, Tiago Neves, Marcelo Parreira do Amaral

978 3 030 94212 0.pdf:
  Titel: Quantifying Quality of Life
  Autor: Katarzyna Wac, Sharon Wulfovich

978 3 030 89147 3.pdf:
  Titel: Learning to Diagnose with Simulations
  Autor: Frank Fischer, Ansgar Opitz

10.1515_9781501764158.pdf:
  Titel: EMPIRE'S VIOLENT END: Comparing Dutch, British, and French Wars of Decolonization, 1945 1962
  Autor: Thijs Brocades Zaalberg, Bart Luttikhuis

978 1 4842 7777 5.pdf:
  Titel: Machine Learning with PySpark
  Autor: Pramod Singh

978 3 030 85679 3.pdf:
  Titel: Demographic and Family Transition in Southeast Asia
  Autor: Wei Jun Jean Yeung

978 3 030 89858 8.pdf:
  Titel: Feeling Political
  Autor: Ute Frevert, Kerstin Maria Pahl, Francesco Buscemi, Philipp Nielsen, Agnes Arndt, Michael Amico

978 3 031 04812 8.pdf:
  Titel: Advances in Computer Science for Engineering and Education
  Autor: Zhengbing Hu, Ivan Dychka, Sergey Petoukhov, Matthew He

978 3 030 93254 1.pdf:
  Titel: Transformation Literacy
  Autor: Petra Künkel, Kristin Vala Ragnarsdottir

978 3 030 95088 0.pdf:
  Titel: A Generalization of Bohr Mollerup's Theorem for Higher Order Convex Functions
  Autor: Jean Luc Marichal, Naïm Zenaïdi

978 3 658 35057 4.pdf:
  Titel: Investition und Finanzierung
  Autor: Hans Paul Becker, Arno Peppmeier

978 3 658 38339 8.pdf:
  Titel: Basiswissen Marketing
  Autor: Gerd Inno Spindler
```

Quelltext A.2: Metadaten der Bücher aus dem Jahr 2022

A.2 Zeitschriften

```
Blockchain Aided Secure Semantic Communication for AI Generated Content in Metaverse.pdf:
  Titel: Blockchain Aided Secure Semantic Communication for AI Generated Content in Metaverse
  Autor: Yijing Lin, Hongyang Du, Dust Niyato, Jiayi Zhang

Fusing Blockchain and AI With Metaverse A Survey.pdf:
  Titel: Fusing Blockchain and AI With Metaverse: A Survey
  Autor: Qinglin Yang, Yetong Zhao, Huawei Huang, Jiawen Kang

When Digital Economy Meets Web3.0 Applications and Challenges.pdf:
  Titel: When Digital Economy Meets Web3.0: Applications and Challenges
  Autor: Chuan Chen, Lei Zhang, Yihao Li, Huawei Huang

Robust_Network_Intrusion_Detection_Through_Explainable_Artificial_Intelligence_XAI.pdf:
  Titel: Robust Network Intrusion Detection Through Explainable Artificial Intelligence (XAI)
  Autor: Pieter Barnard, Nicola Marchetti, Senior Member, Luiz A. DaSilva

AI Enabled_Blockchain_Consensus_Node_Selection_in_Cluster Based_Vehicular_Networks.pdf:
  Titel: AI Enabled Blockchain Consensus Node Selection in Cluster Based Vehicular Networks
  Autor: Khalil Saadat, Ning Wang, Rahim Tafazolli

Cross Atlantic_Experiments_on_EU US_Test Beds.pdf:
  Titel: Cross Atlantic Experiments on EU US Test Beds
  Autor: Sachin Sharma, Avishek Nag, Senior Member, Byrav Ramamurthy

Intelligent_Dynamic_Indoor_Aerosol_Sensing_Using_Terahertz_Band_Wireless_Communication_Systems.pdf:
  Titel: Intelligent Dynamic Indoor Aerosol Sensing Using Terahertz Band Wireless Communication Systems
  Autor: Harun Šiljak, Michael Taynnan Barros, Leo Cooke, Nicola Marchetti

Safety_Score_as_an_Evaluation_Metric_for_Machine_Learning_Models_of_Security_Applications.pdf:
  Titel: Safety Score as an Evaluation Metric for Machine Learning Models of Security Applications
  Autor: Tara Salman, Ali Ghubaish, Devrim Unal, Raj Jain

User_and_Content_Dynamics_of_Edge Aided_Immersive_Reality_Services.pdf:
  Titel: User and Content Dynamics of Edge Aided Immersive Reality Services
  Autor: Olga Chukhno, Olga Galinina, Sergey Andreev, Antonella Molinaro, Antonio Iera

1 s2.0 S2590148619300056 main.pdf:
  Titel: A kernel for multi parameter persistent homology
  Autor: René Corbet, Ulderico Fugacci, Michael Kerber, Claudia Landi, Bei Wang

6G_Perspective_of_Mobile_Network_Operators_Manufacturers_and_Verticals.pdf:
  Titel: 6G Perspective of Mobile Network Operators, Manufacturers, and Verticals
  Autor: René Corbeta, Ulderico Fugaccia, Michael Kerbera, Claudia Landib, Bei Wangc

1 s2.0 S2590148619300068 main.pdf:
  Titel: Representation of NURBS surfaces by Controlled Iterated Functions System automata
  Autor: Lucas Morlet, Christian Gentil, Sandrine Lanquetin, Marc Neveu, Jean Luc Baril

1 s2.0 S2590148619300111 main.pdf:
  Titel: Real time neural network prediction for handling two hands mutual occlusions
  Autor: Dario Pavllo, Mathias Delahaye, Thibault Porssut, Bruno Herbelin, Ronan Boulic

1 s2.0 S2590188520300123 main.pdf:
  Titel: A review on deep learning methods for ECG arrhythmia classification
  Autor: Zahra Ebrahimia, Mohammad Lonib, Masoud Daneshtalabb, Arash Gharehbaghib

Generating_Role Playing_Game_Quests_With_GPT_Language_Models.pdf:
  Titel: Generating Role Playing Game Quests With GPT Language Models
  Autor: Susanna Värtinen, Perttu Hämäläinen, Christian Guckelsberger

Procedural_Puzzle_Generation_A_Survey.pdf:
  Titel: Procedural Puzzle Generation: A Survey
  Autor: Barbara De Kegel, Mads Haahr

Profit_Optimizing_Churn_Prediction_for_Long Term_Loyal_Customers_in_Online_Games.pdf:
  Titel: Profit Optimizing Churn Prediction for Long Term Loyal Customers in Online Games
  Autor: Eunjo Lee, Boram Kim, Sungwook Kang, Byungsoo Kang, Yoonjae Jang, Huy Kang Kim

Reinforcement_Learning_With_Dual Observation_for_General_Video_Game_Playing.pdf:
  Titel: Reinforcement Learning With Dual Observation for General Video Game Playing
  Autor: Chengpeng Hu, Ziqi Wang, Tianye Shu, Hao Tong, Julian Togelius

Win_Prediction_in_Multiplayer_Esports_Live_Professional_Match_Prediction.pdf:
  Titel: Win Prediction in Multiplayer Esports: Live Professional Match Prediction
  Autor: Victoria J. Hodge, Sam Devlin, Nick Sephton, Florian Block

1 s2.0 S2590188520300196 main.pdf:
  Titel: GIMO: A multi objective anytime rule mining system to ease iterative feedback from domain experts
  Autor: Tobias Baum, Steffen Herbold, Kurt Schneider
```

Quelltext A.3: Metadaten der Zeitschriften

A.3 Artikel

```
annurev psych 010418 102744.pdf:
  Titel: Computer Games in Education
  Autor: Richard E. Mayer

41467_2019_Article_13534.pdf:
  Titel: Benchmarking an 11 qubit quantum computer
  Autor: K. Wright

medi 100 e27452.pdf:
  Titel: Reconstruction for diverse fronto orbital defects with computer assisted designed and computer assisted manufactured
       PEEK implants in one stage operation
  Autor: Min Yang, Zhangyi Wu, Hai Yu, Jun Cheng

Lotte_2018_J._Neural_Eng._15_031005.pdf:
  Titel: A review of classification algorithms for EEG based brain computer interfaces: a 10 year update
  Autor: F Lotte, L Bougrain, A Cichocki, M Clerc, M Congedo, A Rakotomamonjy, F Yger

medi 99 e20634.pdf:
  Titel: Computer aided diagnosis system of thyroid nodules ultrasonography
  Autor: Tingting Li, Zirui Jiangb, Man Lu, Shibin Zou

s41095 022 0271 y.pdf:
  Titel: Attention mechanisms in computer vision: A survey
  Autor: Meng Hao Guo, Tian Xing Xu, Jiang Jiang Liu, Zheng Ning Liu

1 s2.0 S0550321323002195 main.pdf:
  Titel: W boson mass and grand unification via the type II seesaw like mechanism
  Autor: Yusuke Shimizu, Shonosuke Takeshita

1 s2.0 S0550321323002225 main.pdf:
  Titel: On the Higgs spectra of the 3 3 1 model with the sextet of scalars engendering the type II seesaw mechanism
  Autor: João Paulo Pinheiro

1 s2.0 S0550321323002353 main.pdf:
  Titel: Muon anomalous magnetic dipole moment in a low scale type I see saw model
  Autor: D.N. Dinh

1 s2.0 S0550321323002158 main.pdf:
  Titel: Comments on ABJM free energy on S3 at large N and perturbative expansions in M theory and string theory
  Autor: M. Beccaria

1 s2.0 S0550321323002183 main.pdf:
  Titel: Nonminimally coupled warm Higgs inflation: Metric vs. Palatini formulations
  Autor: Thammarong Eadkhong

1 s2.0 S0550321323002171 main.pdf:
  Titel: Feshbach Villars oscillator in Kaluza Klein theory
  Autor: Abdelmalek Bouzenada

1 s2.0 S055032132300216X main.pdf:
  Titel: Thermal stability and tunneling radiation in Van der Waals black hole
  Autor: Allah Ditta

1 s2.0 S0550321323002304 main.pdf:
  Titel: Defect localized entropy: Renormalization group and holography
  Autor: Ma Ke Yuan

Journal of Animal Ecology 2017 Weinstein A computer vision for animal ecology.pdf:
  Titel: A computer vision for animal ecology
  Autor: Ben G. Weinstein

EMBR 19 e46628.pdf:
  Titel: Will biologists become computer scientists?
  Autor: Anne Condon, Hélène Kirchner, Damien Larivière, Wallace Marshall

pone.0204566.pdf:
  Titel: Cortical control of a tablet computer by people with paralysis
  Autor: Paul Nuyujukian, Jose Albites Sanabria, Jad Saab, Chethan Pandarinath, Beata Jarosiewicz, Christine H. Blabe, Brian
       Franco

s42400 019 0038 7.pdf:
  Titel: Survey of intrusion detection systems: techniques, datasets and challenges
  Autor: Ansam Khraisat

Computer_Engineering_Education.pdf:
  Titel: Computer Engineering Education
  Autor: Marilyn Wolf

s41746 020 00376 2.pdf:
  Titel: Deep learning enabled medical computer vision
  Autor: Andre Esteva
```

Quelltext A.4: Metadaten der Artikel

A.4 Hochschulschriften

```
978 3 658 25596 1.pdf:
  Titel: Attributionen in der Mensch Computer Interaktion
  Autor: Adelka Niels

BA_K.Khler_FO16w2 B.pdf:
  Titel: Validierung forensischer Software mittels definiertem Testset
  Autor: Kira Marie Kähler

BA_Kittan_Michael_X Ways_Plug in_BPList_Parser.pdf:
  Titel: Realisierung eines Plug ins für die Forensic Software X Ways Forensics für Apple Konfigurationsdateien
  Autor: Michael Kittan

BA_Rademacher_Bastian.pdf:
  Titel: Anwendung eines Computerspiels zur Erkennung und Prävention von ausgewählten psychischen Belastungen bei Studierenden
  Autor: Bastian Rademacher

Bachelorarbeit_Jenny_Felser.pdf:
  Titel: Entwicklung einer Methode zur Empfehlung von Suchbegriffen und  phrasen im forensischen Kontext
  Autor: Jenny Maria Felser

Bachelorarbeit_NeeleFischer.pdf:
  Titel: Evaluierung forensischer Werkzeuge im Bereich der digitalen Forensik
  Autor: Neele Fischer

Bachelorarbeit_Pallmer_Michael_final.pdf:
  Titel: Forensische Sicherheitsanalyse einer Android Applikation
  Autor: Michael Palmer

Bachelorarbeit_Rico_Ludwig_FO15w3.pdf:
  Titel: IT Forensische Analyse moderner Android Betriebssysteme: Sicherheitsbericht zu Android Trojanern
  Autor: Rico Ludwig

Bachelorarbeit Katharina_Schneider.pdf:
  Titel: Vergleich forensischer Werk  zeuge zur Untersuchung von Imagedateien
  Autor: Katharina Jasmin Schneider

Bachelorarbeit.pdf:
  Titel: Entwicklung eines Workflows zur standardisierten Frisurenerstellung im Bereich der computergestützten forensischen
       Gesichtsweichteilrekonstruktion
  Autor: Anna Magdalena Müller

DiplomarbeitlindnerAndreas.pdf:
  Titel: Entwicklung eines Rettungssystems für Industrierechner mit Intel x86 und AMD x64 Architektur
  Autor: Andreas Lindner

HSMW Thesis Vorlage_Becker.pdf:
  Titel: Detektion von falsch positiven Zeitstempeln in Zeitreihen
  Autor: Sarah Becker

HSMW Thesis Vorlage.pdf:
  Titel: Analyse von Log Artefakten bei Android Devices
  Autor: Katharina Engelhardt

Bachelorarbeit_Teply_Paula.pdf:
  Titel: Integration von Aktivitäts  daten mit GPS Positionen
  Autor: Paula Teply

Bachelorarbeit_Daniel_Paasch_FO16w5_B.pdf:
  Titel: Evaluation von Softwarelösungen zur forensischen Auswertung von Datenträgern
  Autor: Daniel Paasch

Bachelorarbeit_AnjaLorenz_DigitaleDidaktik.pdf:
  Titel: Digitale Didaktik: Übertragung didaktischer Modelle in die digitale Welt am Beispiel des Modding Tutorials von
       Railway Empire
  Autor: Anja Lorenz

Bachelorarbeit_Mayerhofer.pdf:
  Titel: Analyse des Renesas Synergy S7G2 Mikrocontrollers in Bezug auf Sicherheitsmerkmale
  Autor: Cora Mayerhofer

Bachelorarbeit_Norman_Tede_Westphal.pdf:
  Titel: Konfigurieren einer Hardware Plattform zur Analyse eines Android Messaging Dienstes
  Autor: Norman Tede Westphal

Diplomarbeit_Mittmannsgruber.pdf:
  Titel: Elektronische Zeiterfassung mittels Android APP mit Datenbankanbindung über TCP  IP Kommunikation
  Autor: Ing. Günther Mittmannsgruber

EntwicklungeinerPlattform.pdf:
  Titel: Entwicklung einer Plattform zur verhaltensbasierten Detektion von Cyber Angriffen und bösartiger Software gegen
       Systeme und Komponenten industrieller Netzwerke
  Autor: Norman Tede Westphal
```

Quelltext A.5: Metadaten der Hochschulschriften

A.5 Paper

```
A Comparative Pilot Study on ErrPs for Different Usage Conditions of an Exoskeleton with a Mobile EEG Device.pdf:
  Titel: A Comparative Pilot Study on ErrPs for Different Usage Conditions of an Exoskeleton with a Mobile EEG Device
  Autor: Svea Marie Meyer, Ashish Rao Mangalore, Stefan K. Ehrlich, Nicolas Berberich, John Nassour, Gordon Cheng

A Robust Low Cost EEG Motor Imagery Based Brain Computer Interface.pdf:
  Titel: A Robust Low Cost EEG Motor Imagery Based Brain Computer Interface
  Autor: Shivanthan Yohanandan, Filiz Isabell Kiral Kornek, Jianbin Tang, Benjamin Scott Mashford, Umar Asif, Stefan Harrer

CNN based Two Step R Peak Detection Method Combining Segmentation and Regression.pdf:
  Titel: CNN based Two Step R Peak Detection Method: Combining Segmentation and Regression
  Autor: Jaeseong Jang, Seongjae Park, Jin Kook Kim, Junho An, and Sunghoon Jung

Creating Computer Vision Models for Respiratory Status Detection.pdf:
  Titel: Creating Computer Vision Models for Respiratory Status Detection
  Autor: Quan T. Doa, Jamil Chaudrib

Decoding Neural Correlation of Language Specific Imagined Speech using EEG Signals.pdf:
  Titel: Decoding Neural Correlation of Language Specific Imagined Speech using EEG Signals
  Autor: Keon Woo Lee, Dae Hyeok Lee, Sung Jin Kim, Seong Whan Lee

Deep Convolutional Neural Network Applied to Electroencephalography Raw Data vs Spectral Features.pdf:
  Titel: Deep Convolutional Neural Network Applied to Electroencephalography: Raw Data vs Spectral Features
  Autor: Dung Truong, Michael Milham, Scott Makeig, Arnaud Delorme

Demonstrating the Viability of Mapping Deep Learning Based EEG Decoders to Spiking Networks on Low powered Neuromorphic Chips.
    pdf:
  Titel: Demonstrating the Viability of Mapping Deep Learning Based EEG Decoders to Spiking Networks on Low Powered
      Neuromorphic Chips
  Autor: Matthijs Pals, Rafael Javier Pérez Belizón, Nicolas Berberich, Stefan K. Ehrlich, John Nassour, Gordon Cheng

Discrimination of Two Class Motor Imagery in a fNIRS Based Brain Computer Interface.pdf:
  Titel: Discrimination of Two Class Motor Imagery in a fNIRS Based Brain Computer Interface
  Autor: Amir H. Moslehi, Mina Bagheri, Anne Marie Ludwig, T. Claire Davies

Mental arithmetic task classification with convolutional neural network based on spectral temporal features from EEG.pdf:
  Titel: Mental arithmetic task classification with convolutional neural network based on spectral temporal features from EEG
  Autor: Zaineb Ajra, Binbin Xu, Gérard Dray, Jacky Montmain, Stephane Perrey

Prototype based Domain Generalization Framework for Subject Independent Brain Computer Interfaces.pdf:
  Titel: Prototype based Domain Generalization Framework for Subject Independent Brain Computer Interfaces
  Autor: Serkan Musellim, Dong Kyun Han, Ji Hoon Jeong, Seong Whan Lee

Teaching Computer Science Students to Communicate Scientific Findings More Effectively.pdf:
  Titel: Teaching Computer Science Students to Communicate Scientific Findings More Effectively
  Autor: Marvin Wyrich, Stefan Wagner

A General purpose Parallel and Heterogeneous Task Programming System for VLSI CAD.pdf:
  Titel: A General purpose Parallel and Heterogeneous Task Programming System for VLSI CAD
  Autor: Tsung Wei Huang

Efficient and Accurate Computational Model of Neuron with Spike Frequency Adaptation.pdf:
  Titel: Efficient and Accurate Computational Model of Neuron with Spike Frequency Adaptation
  Autor: Zubayer Ibne Ferdous, Anlan Yu, Yuan Zeng, Xiaochen Guo, Zhiyuan Yan, Yevgeny Berdichevsky

Latent Space Learning and Feature Learning using Multi template for Multi classification of Alzheimer s Disease.pdf:
  Titel: Latent Space Learning and Feature Learning using Multi template for Multi classification of Alzheimer's Disease
  Autor: Zihao Chen, Haijun Lei, Zhongwei Huang, Baiying Lei

Multi modal Broad Learning System for Medical Image and Text based Classification.pdf:
  Titel: Multi Modal Broad Learning System for Medical Image and Text Based Classification
  Autor: Yanhong Zhou, Jie Du, kai Guan, Tianfu Wang

Project Sized Scaffolding for Software Engineering Courses.pdf:
  Titel: Project Sized Scaffolding for Software Engineering Courses
  Autor: David C. Shepherd, Felipe Fronchetti, Yu Liu, Daqing Hou, Jan DeWaters, Mary Margaret Small

Recognizing Magnification Levels in Microscopic Snapshots.pdf:
  Titel: Recognizing Magnification Levels in Microscopic Snapshots
  Autor: Manit Zaveri, Shivam Kalra, Morteza Babaie, Sultaan Shah

Repairing Brain Computer Interfaces with Fault Based Data Acquisition.pdf:
  Titel: Repairing Brain Computer Interfaces with Fault Based Data Acquisition
  Autor: Cailin Winston, Cleah Winston, Chloe N Winston, Rajesh P N Rao, René Just

System Identification of Decision Making Process in Gold Trading Game.pdf:
  Titel: System Identification of Decision Making Process in Gold Trading Game
  Autor: Mana Yabuki, Tomohiko Utsuki

Towards Naturalistic Speech Decoding from Intracranial Brain Data.pdf:
  Titel: Towards Naturalistic Speech Decoding from Intracranial Brain Data
  Autor: Julia Berezutskaya, Luca Ambrogioni, Nick F. Ramsey, Marcel A.J. van Gerven
```

Quelltext A.6: Metadaten der Paper

Anhang B: Hashwerte

B.1 Bücher

```
978-3-030-51237-8_Datei1.pdf        1453b...1474f
978-3-030-51237-8_Datei2.pdf        1453b...1474f    Gleich

978-3-030-49679-1_Datei1.pdf        debd0...d0792
978-3-030-49679-1_Datei2.pdf        debd0...d0792    Gleich

978-3-030-54871-1_Datei1.pdf        2ddd0...cabf8
978-3-030-54871-1_Datei2.pdf        2ddd0...cabf8    Gleich

978-3-658-31994-6_Datei1.pdf        d71d4...d3a0d
978-3-658-31994-6_Datei2.pdf        d71d4...d3a0d    Gleich

978-1-4614-5632-2_Datei1.pdf        05e42...f0c98
978-1-4614-5632-2_Datei2.pdf        05e42...f0c98    Gleich

978-3-030-66303-2_Datei1.pdf        e0d78...bd07b
978-3-030-66303-2_Datei2.pdf        e0d78...bd07b    Gleich

978-1-4614-4924-9_Datei1.pdf        ccd7e...16c24
978-1-4614-4924-9_Datei2.pdf        ccd7e...16c24    Gleich

978-3-662-08542-4_Datei1.pdf        eb87c...54419
978-3-662-08542-4_Datei2.pdf        eb87c...54419    Gleich

978-3-319-57846-0_Datei1.pdf        41b02...f6b6a
978-3-319-57846-0_Datei2.pdf        41b02...f6b6a    Gleich

978-3-030-36271-3_Datei1.pdf        863ca...a387f
978-3-030-36271-3_Datei2.pdf        863ca...a387f    Gleich

978-1-4020-5604-8_Datei1.pdf        d01a5...625eb
978-1-4020-5604-8_Datei2.pdf        d01a5...625eb    Gleich

978-3-030-45559-0_Datei1.pdf        a3dd1...9fac9
978-3-030-45559-0_Datei2.pdf        a3dd1...9fac9    Gleich

978-3-642-00150-5_Datei1.pdf        3365a...bfa17
978-3-642-00150-5_Datei2.pdf        3365a...bfa17    Gleich

978-3-642-02608-9_Datei1.pdf        e8cca...1de76
978-3-642-02608-9_Datei2.pdf        e8cca...1de76    Gleich

978-3-030-41694-2_Datei1.pdf        afd68...4f37c
978-3-030-41694-2_Datei2.pdf        afd68...4f37c    Gleich

978-3-642-17005-8_Datei1.pdf        30301...e3931
978-3-642-17005-8_Datei2.pdf        30301...e3931    Gleich

978-981-15-6540-3_Datei1.pdf        d61cc...27f48
978-981-15-6540-3_Datei2.pdf        d61cc...27f48    Gleich

978-981-15-7865-6_Datei1.pdf        0025a...c02c7
978-981-15-7865-6_Datei2.pdf        0025a...c02c7    Gleich

978-3-030-37177-7_Datei1.pdf        dd705...2fbac
978-3-030-37177-7_Datei2.pdf        dd705...2fbac    Gleich

978-981-15-0364-1_Datei1.pdf        d26b5...59adb
978-981-15-0364-1_Datei2.pdf        d26b5...59adb    Gleich
```

Quelltext B.1: Hashwerte Springer-Verlag

```
978-1-4842-7777-5_Datei1.pdf          29a47...6b514
978-1-4842-7777-5_Datei2.pdf          29a47...6b514   Gleich

978-3-030-85679-3_Datei1.pdf          09ab2...86c7a
978-3-030-85679-3_Datei2.pdf          09ab2...86c7a   Gleich

978-3-030-94212-0_Datei1.pdf          0b86f...fa5a7
978-3-030-94212-0_Datei2.pdf          0b86f...fa5a7   Gleich

978-3-658-37005-3_Datei1.pdf          d6f0b...704ec
978-3-658-37005-3_Datei2.pdf          d6f0b...704ec   Gleich

978-3-658-28253-0_Datei1.pdf          49862...bd373
978-3-658-28253-0_Datei2.pdf          49862...bd373   Gleich

978-3-031-04812-8_Datei1.pdf          61437...6e41f
978-3-031-04812-8_Datei2.pdf          61437...6e41f   Gleich

978-3-030-89147-3_Datei1.pdf          3b294...3d68c
978-3-030-89147-3_Datei2.pdf          3b294...3d68c   Gleich

978-3-030-95088-0_Datei1.pdf          49265...da4a7
978-3-030-95088-0_Datei2.pdf          49265...da4a7   Gleich

978-981-19-3763-7_Datei1.pdf          112e2...15bb6
978-981-19-3763-7_Datei2.pdf          112e2...15bb6   Gleich

10.1515_9781501764158_Datei1.pdf      e1432...62891
10.1515_9781501764158_Datei2.pdf      e1432...62891   Gleich

978-3-030-96454-2_Datei1.pdf          7e602...3b574
978-3-030-96454-2_Datei2.pdf          7e602...3b574   Gleich

978-3-658-38163-9_Datei1.pdf          c5a28...0fdc7
978-3-658-38163-9_Datei2.pdf          c5a28...0fdc7   Gleich

978-3-658-35057-4_Datei1.pdf          c37e0...d5ada
978-3-658-35057-4_Datei2.pdf          c37e0...d5ada   Gleich

978-3-030-89858-8_Datei1.pdf          eae44...79037
978-3-030-89858-8_Datei2.pdf          eae44...79037   Gleich

978-3-030-93254-1_Datei1.pdf          f6e5f...38b77
978-3-030-93254-1_Datei2.pdf          f6e5f...38b77   Gleich

978-3-658-34970-7_Datei1.pdf          c211d...2277e
978-3-658-34970-7_Datei2.pdf          c211d...2277e   Gleich

978-3-658-36403-8_Datei1.pdf          e84f7...89444
978-3-658-36403-8_Datei2.pdf          e84f7...89444   Gleich

978-3-658-35995-9_Datei1.pdf          edafe...6d4bf
978-3-658-35995-9_Datei2.pdf          edafe...6d4bf   Gleich

978-3-658-38339-8_Datei1.pdf          31409...b5007
978-3-658-38339-8_Datei2.pdf          31409...b5007   Gleich

978-3-662-64435-5_Datei1.pdf          2308f...6997d
978-3-662-64435-5_Datei2.pdf          2308f...6997d   Gleich
```

Quelltext B.2: Hashwerte Jahr 2022

B.2 Zeitschriften

1-s2.0-S2590148619300056-main_Datei1.pdf	02f2b...e64da	
1-s2.0-S2590148619300056-main_Datei2.pdf	dd5f8...5e710	Nicht gleich
1-s2.0-S2590148619300068-main_Datei1.pdf	3c2f1...e1d57	
1-s2.0-S2590148619300068-main_Datei2.pdf	a9632...5e871	Nicht gleich
1-s2.0-S2590148619300111-main_Datei1.pdf	f6388...b22a6	
1-s2.0-S2590148619300111-main_Datei2.pdf	06818...a7766	Nicht gleich
1-s2.0-S2590188520300123-main_Datei1.pdf	432ec...f5dd8	
1-s2.0-S2590188520300123-main_Datei2.pdf	54112...5ef27	Nicht gleich
1-s2.0-S2590188520300196-main_Datei1.pdf	82d92...e5a2f	
1-s2.0-S2590188520300196-main_Datei2.pdf	3f461...dd2ec	Nicht gleich
6G_Perspective_of_Mobile_Network_Operators_Datei1.pdf	57f65...117e5	
6G_Perspective_of_Mobile_Network_Operators_Datei2.pdf	57f65...117e5	Gleich
AI-Enabled_Blockchain_Consensus_Datei1.pdf	fb69e...a4043	
AI-Enabled_Blockchain_Consensus_Datei2.pdf	fb69e...a4043	Gleich
Blockchain-Aided Secure Semantic Communication_Datei1.pdf	1140f...b771e	
Blockchain-Aided Secure Semantic Communication_Datei2.pdf	1140f...b771e	Gleich
Cross-Atlantic_Experiments_on_EU-US_Test-Beds_Datei1.pdf	d43e5...7d816	
Cross-Atlantic_Experiments_on_EU-US_Test-Beds_Datei2.pdf	d43e5...7d816	Gleich
Fusing Blockchain and AI With Metaverse A Survey_Datei1.pdf	3b80c...2a24d	
Fusing Blockchain and AI With Metaverse A Survey_Datei2.pdf	3b80c...2a24d	Gleich
Generating_Role-Playing_Game_Datei1.pdf	66afa...57f74	
Generating_Role-Playing_Game_Datei2.pdf	66afa...57f74	Gleich
Intelligent_Dynamic_Indoor_Aerosol_Sensing_Datei1.pdf	a639f...d0303	
Intelligent_Dynamic_Indoor_Aerosol_Sensing_Datei2.pdf	a639f...d0303	Gleich
Procedural_Puzzle_Generation_A_Survey_Datei1.pdf	115ae...30e05	
Procedural_Puzzle_Generation_A_Survey_Datei2.pdf	115ae...30e05	Gleich
Profit_Optimizing_Churn_Prediction_Datei1.pdf	01e7d...142d5	
Profit_Optimizing_Churn_Prediction_Datei2.pdf	01e7d...142d5	Gleich
Reinforcement_Learning_With_Dual-Observation_Datei1.pdf	c4134...13564	
Reinforcement_Learning_With_Dual-Observation_Datei2.pdf	c4134...13564	Gleich
Robust_Network_Intrusion_Detection_Datei1.pdf	7bca6...1375e	
Robust_Network_Intrusion_Detection_Datei2.pdf	607a4...41b8e	Nicht gleich
Safety_Score_as_an_Evaluation_Metric_Datei1.pdf	33c90...415ea	
Safety_Score_as_an_Evaluation_Metric_Datei2.pdf	33c90...415ea	Gleich
User_and_Content_Dynamics_of_Edge-Aided_Datei1.pdf	83489...83917	
User_and_Content_Dynamics_of_Edge-Aided_Datei2.pdf	83489...83917	Gleich
When Digital Economy Meets Web3.0 Applications_Datei1.pdf	5f6e0...c2d1f	
When Digital Economy Meets Web3.0 Applications_Datei2.pdf	5f6e0...c2d1f	Gleich
Win_Prediction_in_Multiplayer_Esports_Datei1.pdf	bb222...e0a44	
Win_Prediction_in_Multiplayer_Esports_Datei2.pdf	bb222...e0a44	Gleich

Quelltext B.3: Hashwerte Zeitschriften

B.3 Artikel

```
1-s2.0-S055032132300216X-main_Datei1.pdf                977c9...2476a
1-s2.0-S055032132300216X-main_Datei2.pdf                7e96d...16da0   Nicht gleich

1-s2.0-S0550321323002158-main_Datei1.pdf                3bace...a8eba
1-s2.0-S0550321323002158-main_Datei2.pdf                fc2fa...7b567   Nicht gleich

1-s2.0-S0550321323002171-main_Datei1.pdf                ac3e6...807cc
1-s2.0-S0550321323002171-main_Datei2.pdf                16984...3da3b   Nicht gleich

1-s2.0-S0550321323002183-main_Datei1.pdf                db18f...0e3de
1-s2.0-S0550321323002183-main_Datei2.pdf                4591e...ef194   Nicht gleich

1-s2.0-S0550321323002195-main_Datei1.pdf                26ddd...bb953
1-s2.0-S0550321323002195-main_Datei2.pdf                67535...06a11   Nicht gleich

1-s2.0-S0550321323002225-main_Datei1.pdf                a055e...aefdb
1-s2.0-S0550321323002225-main_Datei2.pdf                2670a...82a0f   Nicht gleich

1-s2.0-S0550321323002304-main_Datei1.pdf                e9f2e...964b3
1-s2.0-S0550321323002304-main_Datei2.pdf                b90b4...f4db6   Nicht gleich

1-s2.0-S0550321323002353-main_Datei1.pdf                3d72f...d592c
1-s2.0-S0550321323002353-main_Datei2.pdf                2c4d8...4fa20   Nicht gleich

41467_2019_Article_13534_Datei1.pdf                     b2192...e5491
41467_2019_Article_13534_Datei2.pdf                     b2192...e5491   Gleich

annurev-psych-010418-102744_Datei1.pdf                  6ecb8...066ae
annurev-psych-010418-102744_Datei2.pdf                  dc07c...350fc   Nicht gleich

Computer_Engineering_Education_Datei1.pdf               31616...f120f
Computer_Engineering_Education_Datei2.pdf               31616...f120f   Gleich

EMBR-19-e46628_Datei1.pdf                               9b65c...d5033
EMBR-19-e46628_Datei2.pdf                               9b65c...d5033   Gleich

A computer vision for animal ecology_Datei1.pdf         a0f66...de733
A computer vision for animal ecology_Datei2.pdf         d36e9...1bc85   Nicht gleich

Lotte_2018_J._Neural_Eng._15_031005_Datei1.pdf          80dcb...59da9
Lotte_2018_J._Neural_Eng._15_031005_Datei2.pdf          b60a9...81fd6   Nicht gleich

medi-100-e27452_Datei1.pdf                              74858...4f13e
medi-100-e27452_Datei2.pdf                              74858...4f13e   Gleich

medi-99-e20634_Datei1.pdf                               66cb5...bbfd3
medi-99-e20634_Datei2.pdf                               66cb5...bbfd3   Gleich

pone.0204566_Datei1.pdf                                 eb410...94e19
pone.0204566_Datei2.pdf                                 eb410...94e19   Gleich

s42400-019-0038-7_Datei1.pdf                            28bca...e080a
s42400-019-0038-7_Datei2.pdf                            28bca...e080a   Gleich

s41746-020-00376-2_Datei1.pdf                           4d421...69bfb
s41746-020-00376-2_Datei2.pdf                           4d421...69bfb   Gleich

s41095-022-0271-y_Datei1.pdf                            56bff...23768
s41095-022-0271-y_Datei2.pdf                            56bff...23768   Gleich
```

Quelltext B.4: Hashwerte Artikel

B.4 Hochschulschriften

```
Bachelorarbeit_Norman_Tede_Westphal_Datei1.pdf                    1481a...ddc03
Bachelorarbeit_Norman_Tede_Westphal_Datei2.pdf                    1481a...ddc03    Gleich

Bachelorarbeit_NeeleFischer_Datei1.pdf                           2d800...1ff04
Bachelorarbeit_NeeleFischer_Datei2.pdf                           2d800...1ff04    Gleich

Bachelorarbeit_Mayerhofer_Datei1.pdf                             1ee6a...36702
Bachelorarbeit_Mayerhofer_Datei2.pdf                             1ee6a...36702    Gleich

Bachelorarbeit_Datei1.pdf                                        9d328...47dfa
Bachelorarbeit_Datei2.pdf                                        9d328...47dfa    Gleich

978-3-658-25596-1_Datei1.pdf                                     6d089...1549d
978-3-658-25596-1_Datei2.pdf                                     6d089...1549d    Gleich

Bachelorarbeit_Jenny_Felser_Datei1.pdf                           538c0...46c09
Bachelorarbeit_Jenny_Felser_Datei2.pdf                           538c0...46c09    Gleich

BA_Rademacher_Bastian_Datei1.pdf                                 a8487...86c84
BA_Rademacher_Bastian_Datei2.pdf                                 a8487...86c84    Gleich

Bachelorarbeit-Katharina_Schneider_Datei1.pdf                    fd139...af2f0
Bachelorarbeit-Katharina_Schneider_Datei2.pdf                    fd139...af2f0    Gleich

Diplomarbeit_Mittmannsgruber_Datei1.pdf                          b1ad4...e7ede
Diplomarbeit_Mittmannsgruber_Datei2.pdf                          b1ad4...e7ede    Gleich

Bachelorarbeit_Pallmer_Michael_final_Datei1.pdf                  0d89a...881bb
Bachelorarbeit_Pallmer_Michael_final_Datei2.pdf                  0d89a...881bb    Gleich

Bachelorarbeit_AnjaLorenz_DigitaleDidaktik_Datei1.pdf            6575c...a4bbd
Bachelorarbeit_AnjaLorenz_DigitaleDidaktik_Datei2.pdf            6575c...a4bbd    Gleich

BA_K.Khler_FO16w2-B_Datei1.pdf                                   6ca31...b1c35
BA_K.Khler_FO16w2-B_Datei2.pdf                                   6ca31...b1c35    Gleich

HSMW-Thesis-Vorlage_Datei1.pdf                                   f3d72...fd8fa
HSMW-Thesis-Vorlage_Datei2.pdf                                   f3d72...fd8fa    Gleich

Bachelorarbeit_Teply_Paula_Datei1.pdf                            e7f27...ab4cc
Bachelorarbeit_Teply_Paula_Datei2.pdf                            e7f27...ab4cc    Gleich

Bachelorarbeit_Daniel_Paasch_FO16w5_B_Datei1.pdf                 fd9be...7d8f5
Bachelorarbeit_Daniel_Paasch_FO16w5_B_Datei2.pdf                 fd9be...7d8f5    Gleich

HSMW-Thesis-Vorlage_Becker_Datei1.pdf                            ca728...d4c59
HSMW-Thesis-Vorlage_Becker_Datei2.pdf                            ca728...d4c59    Gleich

BA_Kittan_Michael_X-Ways_Plug-in_BPList_Parser_Datei1.pdf        b105b...30342
BA_Kittan_Michael_X-Ways_Plug-in_BPList_Parser_Datei2.pdf        b105b...30342    Gleich

DiplomarbeitlindnerAndreas_Datei1.pdf                            7429b...fee9c
DiplomarbeitlindnerAndreas_Datei2.pdf                            7429b...fee9c    Gleich

EntwicklungeinerPlattform_Datei1.pdf                             f6185...11d69
EntwicklungeinerPlattform_Datei2.pdf                             f6185...11d69    Gleich

Bachelorarbeit_Rico_Ludwig_FO15w3_Datei1.pdf                     2314b...78d18
Bachelorarbeit_Rico_Ludwig_FO15w3_Datei2.pdf                     2314b...78d18    Gleich
```

Quelltext B.5: Hashwerte Hochschulschriften

B.5 Paper

```
A Comparative Pilot Study on ErrPs_Datei1.pdf                    31651...c0dfb
A Comparative Pilot Study on ErrPs_Datei2.pdf                    31651...c0dfb   Gleich

A General-purpose Parallel and Heterogeneous Task_Datei1.pdf     bbb3b...57b91
A General-purpose Parallel and Heterogeneous Task_Datei2.pdf     bbb3b...57b91   Gleich

A Robust Low-Cost EEG Motor_Datei1.pdf                           1d1bc...31eb6
A Robust Low-Cost EEG Motor_Datei2.pdf                           1d1bc...31eb6   Gleich

CNN-based Two Step R Peak Detection Method_Datei1.pdf            f9a51...6ecc6
CNN-based Two Step R Peak Detection Method_Datei2.pdf            f9a51...6ecc6   Gleich

Creating Computer Vision Models_Datei1.pdf                       0134f...dac7d
Creating Computer Vision Models_Datei2.pdf                       0134f...dac7d   Gleich

Decoding Neural Correlation of Language-Specific_Datei1.pdf      6eb51...bf136
Decoding Neural Correlation of Language-Specific_Datei2.pdf      6eb51...bf136   Gleich

Deep Convolutional Neural Network_Datei1.pdf                     4b5fb...29999
Deep Convolutional Neural Network_Datei2.pdf                     4b5fb...29999   Gleich

Demonstrating the Viability of Mapping Deep Learning_Datei1.pdf  1f67e...bb0ff
Demonstrating the Viability of Mapping Deep Learning_Datei2.pdf  1f67e...bb0ff   Gleich

Discrimination of Two-Class Motor Imagery_Datei1.pdf             a3e02...dce6f
Discrimination of Two-Class Motor Imagery_Datei2.pdf             a3e02...dce6f   Gleich

Efficient and Accurate Computational Model_Datei1.pdf            4240c...64706
Efficient and Accurate Computational Model_Datei2.pdf            4240c...64706   Gleich

Latent Space Learning and Feature Learning_Datei1.pdf            bede9...e3184
Latent Space Learning and Feature Learning_Datei2.pdf            bede9...e3184   Gleich

Mental arithmetic task classification_Datei1.pdf                 279c6...b1588
Mental arithmetic task classification_Datei2.pdf                 279c6...b1588   Gleich

Multi-modal Broad Learning System_Datei1.pdf                     53d64...8e744
Multi-modal Broad Learning System_Datei2.pdf                     53d64...8e744   Gleich

Project-Sized Scaffolding_Datei1.pdf                             7a7c0...61463
Project-Sized Scaffolding_Datei2.pdf                             7a7c0...61463   Gleich

Prototype-based Domain Generalization Framework_Datei1.pdf       90a8d...48fd9
Prototype-based Domain Generalization Framework_Datei2.pdf       90a8d...48fd9   Gleich

Recognizing Magnification Levels_Datei1.pdf                      394a0...b771b
Recognizing Magnification Levels_Datei2.pdf                      394a0...b771b   Gleich

Repairing Brain-Computer Interfaces_Datei1.pdf                   23453...0cfc4
Repairing Brain-Computer Interfaces_Datei2.pdf                   23453...0cfc4   Gleich

System Identification of Decision-Making Process_Datei1.pdf      6b767...35cf9
System Identification of Decision-Making Process_Datei2.pdf      6b767...35cf9   Gleich

Teaching Computer Science Students_Datei1.pdf                    9dc20...e28a7
Teaching Computer Science Students_Datei2.pdf                    9dc20...e28a7   Gleich

Towards Naturalistic Speech Decoding_Datei1.pdf                  a486a...d1f02
Towards Naturalistic Speech Decoding_Datei2.pdf                  a486a...d1f02   Gleich
```

Quelltext B.6: Hashwerte Paper

Literaturverzeichnis

[1] Bundeskriminalamt, *Polizeiliche Kriminalstatistik 2022: T01 Grundtabelle - Fälle ab 1987 (V1.0)*, Online; Accessed: 26. August 2023. Adresse: https://www.bka.de/DE/AktuelleInformationen/StatistikenLagebilder/PolizeilicheKriminalstatistik/PKS2022/PKSTabellen/Zeitreihen/zeitreihen_node.html.

[2] W. Becker, P. Ulrich, O. Schmid und C. Feichtinger, „Grundlagen“, in *Industrielle Digitalisierung: Entwicklungen und Strategien für mittelständische Unternehmen*. Wiesbaden: Springer Fachmedien Wiesbaden, 2020, S. 1, ISBN: 978-3-658-28815-0. DOI: 10.1007/978-3-658-28815-0_3. Adresse: https://doi.org/10.1007/978-3-658-28815-0_3.

[3] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 1–2, ISBN: 3642569714.

[4] J. Eberspächer, *Die Zukunft der Printmedien*, ger, 1st ed. 2002. 2002, S. 200, ISBN: 3642561578.

[5] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 19, ISBN: 3642569714.

[6] A. Gohr, *Dateiaufbau*, Online; Accessed: 9. August 2023. Adresse: http://www.p2501.ch/pdf-howto/grundlagen/aufbau_und_syntax/aufbau.

[7] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 32, ISBN: 3642569714.

[8] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 43–44, ISBN: 3642569714.

[9] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 18, ISBN: 3642569714.

[10] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 25, ISBN: 3642569714.

[11] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 34–36, ISBN: 3642569714.

[12] Z. F. Makhrib und A. A. Karim, „Improved Fragile Watermarking Technique Using Modified LBP Operator“, in *2022 International Conference on Computer Science and Software Engineering (CSASE)*, 2022, S. 132–137. DOI: 10.1109/CSASE51777.2022.9759647.

[13] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 33, ISBN: 3642569714.

[14] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 137, ISBN: 3642569714.

[15] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 27, ISBN: 3642569714.

[16] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 26, ISBN: 3642569714.

[17] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 28, ISBN: 3642569714.

[18] J. Eberspächer, *Die Zukunft der Printmedien*, ger, 1st ed. 2002. 2002, S. 203–204, ISBN: 3642561578.

[19] J. Eberspächer, *Die Zukunft der Printmedien*, ger, 1st ed. 2002. 2002, S. 204, ISBN: 3642561578.

[20] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 29, ISBN: 3642569714.

[21] Y. Wang, D. Gong, B. Lu, F. Xiang und F. Liu, „Exception Handling-Based Dynamic Software Watermarking“, *IEEE Access*, Jg. 6, S. 8882–8889, 2018. DOI: 10.1109/ACCESS.2018.2810058.

[22] J. Eberspächer, *Die Zukunft der Printmedien*, ger, 1st ed. 2002. 2002, S. 205, ISBN: 3642561578.

[23] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 30, ISBN: 3642569714.

[24] A. Samcovic und J. Turán, „Digital image watermarking by spread spectrum“, Jan. 2007.

[25] Wikibrief, *Burst-Übertragung*, Online; Accessed: 26. Juli 2023. Adresse: https://de.wikibrief.org/wiki/Burst_transmission.

[26] K. Dipl.-Ing. Lipinski, *Frequenzsprungverfahren*, Online; Accessed: 26. Juli 2023. Adresse: https://cutt.ly/zwg5ZDp9.

[27] R. Dubolia, R. Singh, S. S. Bhadoria und R. Gupta, „Digital Image Watermarking by Using Discrete Wavelet Transform and Discrete Cosine Transform and Comparison Based on PSNR“, in *2011 International Conference on Communication Systems and Network Technologies*, 2011, S. 593–596. DOI: 10.1109/CSNT.2011.127.

[28] D. Somwanshi, I. Chhipa, T. Singhal und A. Yadav, „Modified Least Significant Bit Algorithm of Digital Watermarking for Information Security“, in Jan. 2018, S. 473–484, ISBN: 978-981-10-5698-7. DOI: 10.1007/978-981-10-5699-4_44.

[29] A. Muzakir und M. Habibi, „Watermarking Techniques Using Least Significant Bit Algorithm for Digital Image Security Standard Solution- Based Android“, *Scientific Journal of Informatics*, Jg. 4, S. 20, Mai 2017. DOI: 10.15294/sji.v4i1.7290.

[30] A. Lang, S. Thiemert, M. Steinebach, J. Dittmann und E. Hauer, „Ausgewählte Angriffe der Stirmark Benchmark Suite“, *Fraunhofer Institut IPSI*, S. 320–332, 2002.

[31] F. A. P. Petitcolas, R. J. Anderson und M. G. Kuhn, „Attacks on copyright marking systems“, S. 224–227, Apr. 1998.

[32] F. A. P. Petitcolas, R. J. Anderson und M. G. Kuhn, „Attacks on copyright marking systems“, S. 228–229, Apr. 1998.

[33] J. Dittmann, *Digitale Wasserzeichen Grundlagen, Verfahren, Anwendungsgebiete* (Xpert.press), ger, 1st ed. 2000. 2000, S. 36–40, ISBN: 3642569714.

[34] S. Dhenakaran und K. T. Sambanthan, „Web crawler-an overview“, *International Journal of Computer Science and Communication*, Jg. 2, Nr. 1, S. 265–267, 2011.

[35] S. Lange und R. Bender, „Das Histogramm“, *DMW-Deutsche Medizinische Wochenschrift*, Jg. 132, Nr. S 01, e7–e8, 2007.

[36] M. Maes, „Twin Peaks: The Histogram Attack to Fixed Depth Image Watermarks“, Online; Accessed: 5. Juli 2023. Adresse: https://users.ece.cmu.edu/~adrian/487-s06/maes-twin-peaks-attack.pdf.

[37] F. Fischer, „Digitale Wasserzeichen“, *Institut für Informatik der Universität Zürich*, 2001, Online; Accessed: 3. Juli 2023. Adresse: http://home.datacomm.ch/felix_fischer/data/Semesterarbeit.pdf.

[38] S. Titscher, R. Wodak, M. Meyer und E. Vetter, „Textanalysemethoden in Kurzdarstellung“, in *Methoden der Textanalyse: Leitfaden und Überblick*. Wiesbaden: VS Verlag für Sozialwissenschaften, 1998, S. 73–218, ISBN: 978-3-322-87302-6. DOI: 10.1007/978-3-322-87302-6_3. Adresse: https://doi.org/10.1007/978-3-322-87302-6_3.

[39] S. Titscher, R. Wodak, M. Meyer und E. Vetter, „Textanalysemethoden in Kurzdarstellung“, in *Methoden der Textanalyse: Leitfaden und Überblick*. Wiesbaden: VS Verlag für Sozialwissenschaften, 1998, S. 73–218, ISBN: 978-3-322-87302-6. DOI: 10.1007/978-3-322-87302-6_3. Adresse: https://doi.org/10.1007/978-3-322-87302-6_3.

[40] H. Kubicek, A. Breiter und J. Jarke, „Daten, metadaten, interoperabilität“, *Handbuch digitalisierung in staat und verwaltung*, S. 1–13, 2019.

[41] M. Mühlhäuser, „Open Metadata: Nutzerzentrierte wettbewerbliche Datenverwertung mit offenen Rahmendaten“, in *Die Zukunft der Datenökonomie: Zwischen Geschäftsmodell, Kollektivgut und Verbraucherschutz*, C. Ochs, M. Friedewald, T. Hess und J. Lamla, Hrsg. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, S. 80, ISBN: 978-3-658-27511-2. DOI: 10.1007/978-3-658-27511-2_5. Adresse: https://doi.org/10.1007/978-3-658-27511-2_5.

[42] M. Mühlhäuser, „Open Metadata: Nutzerzentrierte wettbewerbliche Datenverwertung mit offenen Rahmendaten“, in *Die Zukunft der Datenökonomie: Zwischen Geschäftsmodell, Kollektivgut und Verbraucherschutz*, C. Ochs, M. Friedewald, T. Hess und J. Lamla, Hrsg. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, S. 79, ISBN: 978-3-658-27511-2. DOI: 10.1007/978-3-658-27511-2_5. Adresse: https://doi.org/10.1007/978-3-658-27511-2_5.

[43] M. Mühlhäuser, „Open Metadata: Nutzerzentrierte wettbewerbliche Datenverwertung mit offenen Rahmendaten“, in *Die Zukunft der Datenökonomie: Zwischen Geschäftsmodell, Kollektivgut und Verbraucherschutz*, C. Ochs, M. Friedewald, T. Hess und J. Lamla, Hrsg. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, S. 81, ISBN: 978-3-658-27511-2. DOI: 10.1007/978-3-658-27511-2_5. Adresse: https://doi.org/10.1007/978-3-658-27511-2_5.

[44] J. Buchmann, *Einführung in die Kryptographie* (Springer-Lehrbuch), ger, 6., überarb. Aufl. 2016. 2016, S. 234–243, ISBN: 9783642397752.

[45] J. Pelzl und C. Paar, „Hash-Funktionen“, in *Kryptografie verständlich: Ein Lehrbuch für Studierende und Anwender*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, S. 335–362, ISBN: 978-3-662-49297-0. DOI: 10.1007/978-3-662-49297-0_11. Adresse: https://doi.org/10.1007/978-3-662-49297-0_11.

[46] A. Pratzner, *Python Kurs: Mit Python programmieren lernen für Anfänger und Fortgeschrittene*, Online; Accessed: 18. Juli 2023, 2023. Adresse: https://docs.python.org/3/library/functions.html#open.

[47] *Comparing and Merging Files*, Online; Accessed: 17. August 2023, 2023. Adresse: https://www.gnu.org/software/diffutils/manual/diffutils.html#Output-Formats.

Eidesstattliche Erklärung

Hiermit versichere ich – Raika Käbisch – an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.

Mittweida, 01. September 2023 [redacted]

Ort, Datum Raika Käbisch