Steinebach, Wolf, Hübler; Wasserzeichentreue Bildbearbeitung, Tagungsband D-A-CH Security 2006, Horster (Hrsg.), syssec Verlag, ISBN 3-00-018166-0, S. 248-259, 2006

Wasserzeichentreue Bildbearbeitung

Martin Steinebach, Patrick Wolf, Veronika Hübler

Fraunhofer Institut Integrierte Publikations- und Informationssysteme
Dolivostraße 15; 64293 Darmstadt
(martin.steinebach | patrick.wolf)@ipsi.fraunhofer.de

Zusammenfassung

Digitale Wasserzeichen sind ein Mechanismus zum Schutz digitaler Medien. Mit ihrer Hilfe können unterschiedliche Schutzziele adressiert werden, insbesondere Authentizität, also die vertrauenswürdige Identifikation von Sender (Urheber) oder Empfänger (Kunde) eines Mediums, und Integrität, also den Nachweis der Unversehrtheit des Mediums. Die vorliegende Arbeit stellt ein Konzept vor, wie Bilder, deren Integrität mit digitalen Wasserzeichen geschützt wird, bearbeitet werden können, ohne den eingebetteten Schutz zu zerstören. Wir stellen einen Bildeditor vor, der zum einen Bilder mittels eines Wasserzeichens schützen kann, zum anderen aber auch die Nachbearbeitung erlaubt. So kann beispielsweise ermöglicht werden, dass ein Bild bis zu einem bestimmten Grad komprimiert oder skaliert werden kann, die Bildaussage jedoch durch Abschneiden von Bildbereichen oder retuschieren von Bildelementen nicht verändert werden darf. Des Weiteren sollte die Anwenderfreundlichkeit dadurch gewährleistet sein, dass bereits während der Bearbeitung in einem Bildeditor auf nicht zulässige Bearbeitungsschritte hingewiesen wird.

1 Motivation

Der Betrachter eines digitalen Bildes kann heutzutage nicht mehr davon ausgehen, dass dieses Bild der Realität entspricht und nicht mutwillig manipuliert wurde. Will man Bilder vor Gericht als Beweismittel heranziehen oder handelt es sich um sensible Aufnahmen wie medizinische Fotografien oder militärische Satellitenbilder, wird klar, wie wichtig die Nachweisbarkeit der Integrität – der Unversehrtheit - eines Bildes sein kann.

In dieser Arbeit stellen wir einen Bildeditor vor, der in der Lage ist, die Integrität eines Bildes, das in einem herkömmlichen Bildformat (JPEG, TIFF, etc.) vorliegt, während des gesamten Publikationsprozesses zu gewährleisten. Dazu beschreiben wir zunächst, wie Integrität digitaler Daten im Allgemeinen geschützt wird und welche Besonderheiten der Schutz der Integrität digitaler Medien im Speziellen mit sich bringt. Insbesondere wird dabei auf Inhaltsbeschreibungen über Bildmerkmale und die Eignung digitaler Wasserzeichen zum Integritätsschutz eingegangen. Anschließend wird die Wichtigkeit der einfachen und effektiven Einbindung von Schutzmaßnahmen in Publikationsprozesse diskutiert. Im folgenden Abschnitt wird mit Hilfe der gewonnen Erkenntnisse der wasserzeichentreue Editor entworfen und dessen Implementierung vorgestellt.

1.1 Integritätsschutz digitaler Daten

Um digitale Bilder vor unerlaubten Veränderungen zu schützen, besteht die Möglichkeit, sie einfach als digitale Daten zu betrachten und für Speicherung und Übertragung klassische kryptografische Verfahren einzusetzen. Hierbei unterscheidet man zwischen symmetrischen Verfahren, bei denen der gleiche Schlüssel zum Verschlüsseln verwendet wird wie beim Entschlüsseln der Bilddatei und den asymmetrischen Verfahren, bei denen jeweils verschiedene Schlüssel verwendet werden [B99].

Symmetrische Verschlüsselungsverfahren eignen sich nicht zum Schutz der Integrität eines Bildes, wenn es publiziert werden soll, da nur eine eingeschränkte Anzahl von Personen den Schlüssel kennt und damit das Bild ansehen darf. Aber auch beim Einsatz eines asymmetrischen Verschlüsselungsverfahrens wird die Benutzbarkeit eingeschränkt, da, bevor das Bild angesehen werden kann, der öffentliche Schlüssel (public key) gebraucht wird. Beide Verfahren haben gemein, dass sie somit nur zum Schutz des Transports geeignet sind. Nach der Entschlüsselung kann das Bild frei manipuliert werden.

Ein anderer Ansatz, der auch von Friedman [Fri93] vorgestellt wird, wäre das Bild, möglichst schon bei der Aufnahme mit der digitalen Kamera, mit einer digitalen Signatur zu versehen. Bei diesem Verfahren wird ein Hash, also eine kompakte, eindeutige Beschreibung der Bilddaten erzeugt und dieser mit einem privaten Schlüssel (private key) geschützt. Mit Hilfe eines öffentlichen Schlüssels kann der signierte Hash entschlüsselt werden und mit dem Hash des vorliegenden Bildes verglichen werden, um so die Herkunft des Bildes und die Bildintegrität festzustellen.

All diese Ansätze zeichnen sich durch zwei Kennzeichen aus. Erstens, benötigen sie zum Nachweis der Integrität außer den zu schützenden Daten noch zusätzliche Informationen, die zumeist in separaten Dateien vorliegen. Zweitens, schützen sie nur die *binäre* Integrität, das heißt die bitweise Übereinstimmung der Daten. In vielen Fällen ist es jedoch nicht die binäre Integrität, die geschützt werden soll, sondern die *semantische* Integrität [Ste04]. Sie liegt vor, wenn der *Inhalt* der Datei sich nach einer Operation nicht vom Inhalt der ursprünglichen Daten unterscheidet. Diese Unterscheidung ist notwendig, wenn man inhaltserhaltende Veränderungen an einem Medium zulassen möchte, ohne auf den Integritätsschutz zu verzichten.

Im Publikationsprozess ist es üblich digitale Medien an den Publikationskontext anzupassen, zum Beispiel sind für Bilder auf Webseiten oft fixe Größen vorgesehen, so dass das Bild skaliert werden muss, oder um ein bestimmtes Seitenverhältnis zu erreichen, muss ein (kleiner) Teil vom Rand abgeschnitten werden. Es ist also dadurch natürlich, dass im Publikationsprozess die Daten digitaler Medien verändert werden, weswegen der Schutz der binären Integrität nicht als Integritätsschutz für digitale Medien ausreichen kann.

1.2 Integritätsschutz digitaler Medien

Um die semantische Integrität digitaler Medien zu schützen, gibt es zwei grundsätzliche Ansätze: Zum einen wird versucht, ähnlich wie bei Hash-Verfahren die Daten eines digitalen Mediums erfasst werden, nun den *Inhalt* (die in den Mediendaten enthaltende Information) in Form von Merkmalen zu erfassen. Dabei dürfen sich diese Merkmale nicht durch inhaltserhaltende Manipulationen ändern. Zum anderen kann das Medium mit Informationen angereichert, die es erlauben Manipulationen zu erkennen. Ersterer Ansatz führt zur Extraktion von Bildmerkmalen, letzterer zu digitalen Wasserzeichen.

1.2.1 Bildmerkmale zur Integritätsbeurteilung

Computer können selbst minimale Änderungen in Daten zuverlässig erkennen, aber es ist für sie schwierig Inhalte zu erfassen. Menschen hingegen können (leichte) Änderungen an Daten nicht erkennen (siehe Transparenz digitaler Wasserzeichen), aber dafür können sie den Unterschied zwischen einer inhaltsverändernden und einer inhaltserhaltenden Manipulation im Allgemeinen auf den ersten Blick erkennen, vorausgesetzt ihnen ist das Original bekannt.

Um ein Bild vor inhaltsverändernden Manipulationen zu schützen und inhaltserhaltende Manipulationen zu erlauben, muss definiert werden, was den Inhalt des Bildes ausmacht. Man muss sich von der binären Betrachtungsweise des Bildes distanzieren und eine semantische Herangehensweise an das Bild finden. Dazu ist es wichtig, die Bildmerkmale zu betrachten, die für die menschliche Wahrnehmung als inhaltstragend gelten[1].

In der Literatur, beispielsweise in [D2000], werden hierzu verschiedene Merkmale vorgeschlagen. Die verbreitetsten sind:

- *Bildkanten*: Anhand verschiedener Algorithmen, z.B. [MG01] wird aus einem Bild mit Flächen und Texturen eine reine Kantenabbildung. Diese kann dann zu weiteren Abstraktionen führen. In dem das Bild in einzelne Blöcke zerlegt wird, können Anzahl oder Verlauf der Kanten innerhalb eines Blocks als Merkmalsvektor verwendet werden. Oft werden die Kanten auch mittels eines Schwellwertes bezüglich ihrer Intensität (Schärfe) zwischen relevanten in irrelevanten Kanten unterschieden.
- *Histogramm*: Aus dem Bildmaterial können verschiedene Typen von Histogrammen errechnet werden, welche als beschreibendes Merkmal des Bildes dienen. Dabei können beispielsweise Farb- oder Helligkeitsverteilungen eingesetzt werden. Die Histogramme können sowohl im Bild- als auch im Frequenzbereich gebildet werden und ebenfalls blockweise erstellt werden.
- *Gewichtung*: In [DSF+01] wird eine Triangulierung nach Delaunay eingesetzt. Zu je drei extrahierten Kanten wird der Schwerpunkt des Dreiecks, das ihre Mittelpunkte bilden, berechnet. Der Vorteil liegt darin, damit sehr kompakte Merkmale zu erhalten, die trotzdem eine starke Bindung zum Ausgangsmaterial haben und sensible auf Veränderungen des Bildes reagieren.
- *Fluchtpunkte*: Fluchtpunkte werden ebenfalls aus Kanten gewonnen. Sie sagen etwas über den (scheinbaren) Standpunkt des Betrachters bzw. der Kamera aus.
- *MPEG*-7: Im Standard MPEG-7[2] werden eine große Reihe weiterer beschreibender Merkmale („descriptors“) definiert, welche beispielsweise Gesichtserkennung oder Texturbeschreibungen umfassen. Diese Merkmale können auch im Rahmen der Integritätserkennung eingesetzt werden.

Allerdings ist es sehr anwendungsabhängig, welche Operationen an einem Bild als inhaltsverändernd angesehen werden und welche nicht [DS00]. Als Beispiel werden hier militärische Satellitenaufnahmen angeführt, bei denen bereits Pixelveränderungen durch beispielsweise

[1] Man kann mit Merkmalen alleine nicht die Semantik des Bildes erfassen. Sie helfen nur festzustellen, was sich wahrnehmbar am Bild verändert, was eine notwendige Voraussetzung für Inhaltsveränderungen ist.

[2] http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm

verlustbehaftete Kompression als inhaltsverändernd gelten können. Im Gegensatz hierzu stehen Überwachungsfotos im Straßenverkehr, bei denen der Sachverhalt auch nach Kompression und Skalierung erkannt werden kann.

1.2.2 Integritätsschutz durch Wasserzeichen

Ähnlich wie Wasserzeichen in Papier verwendet werden um die Echtheit, Originalität und die Urheberschaft eines Dokumentes nachzuweisen, wurden digitale Wasserzeichen entwickelt, um diesen Schutzmechanismus auch auf digitale Daten übertragen zu können. Es handelt sich hier um für Menschen nicht wahrnehmbare Muster, welche in das Datenmaterial (z.B. Bild, Audio, Video) mit einem Einbettungsalgorithmus eingebracht werden [Dit00].

Zwei wesentliche Eigenschaften von digitalen Wasserzeichen sind Transparenz und Robustheit. Die Transparenz ist ein Maß für die Wahrnehmbarkeit der Veränderung, die durch das Einbetten von Informationen entsteht. Die Robustheit ist ein Maß dafür, ob und in wie weit die eingebetteten Informationen (auch kurz *das* Wasserzeichen genannt) auch nach Manipulationen des Trägermediums noch auszulesen sind. Dabei reicht die Skala der Robustheit von fragil, das heißt, das Wasserzeichen wird bei Manipulationen vollständig zerstört, bis hin zu robust, was bedeutet, dass die eingebettete Information auch nach der Manipulation des Trägermaterials noch vollständig vorhanden ist. Einige Autoren (vgl. [CMB02]) unterscheiden auch je nach Art der Manipulation am Trägermedium zwischen Robustheit (Veränderung durch übliche Medienverarbeitung) und Sicherheit (gezielte Angriffe gegen das Wasserzeichen).

Ein einfacher Schutz der Integrität ist mit fragilen Wasserzeichen möglich. Das Wasserzeichen wird hier nur mit geringer Robustheit eingebracht. Beim Ausleseprozess wird geprüft, ob die Wasserzeicheninformation noch vorhanden ist. Kann das Wasserzeichen noch ausgelesen werden, so ist die Integrität des Bildes bewiesen. Ein Beispiel hierfür findet man bei Walton [Wal95], der Prüfsummen von signifikanten Bildpixeln mit verschlüsselten, pseudozufälligen Mustern in die LSBs[3] (least significant bits) des Bildes eingebracht hat. Andere Beispiele Pixelblock-basierte Ansätze finden sich in [WD96], [Wal95] und [Fri98]. Neuere Verfahren sind sogar in der Lage, Änderungen am Bild zu lokalisieren [LFS04].

Aber solche Ansätze können auch nicht die semantische Integrität schützten. Verfahren zum Schutz der semantischen Integrität müssen robust gegen bestimmte erlaubte Verarbeitungsschritte sein und fragil gegenüber unerlaubten. Wasserzeichenverfahren, die solche eine Unterscheidung erlauben, bezeichnet man als semi-fragil. Bereits Wolfgang und Delp [WD96] haben statt den üblichen fragilen Schutzverfahren ein Wasserzeichen vorgestellt, das robust gegenüber JPEG-Kompression bis zu einem bestimmten Grad ist, andere Änderungen jedoch nicht lokalisieren kann. Hier werden für die Einbettung der Bildmerkmale in das Original nicht die Werte des Originalbildes verwendet, sondern die Werte des bereits komprimierten Originals. Fridrichs Verfahren [Fri98] dagegen beschränkt sich nicht auf die Robustheit gegen JPEG-Kompression. Beim blockweisen Auslesen seines robusten Wasserzeichens kann die Intensität des Wasserzeichens festgestellt werden. Ist die Intensität der Wasserzeichen in allen Bildbereichen gleich, so handelt es sich wahrscheinlich um zugelassene Änderungen an Bild wie Kompression, Gamma-Korrektur, Skalierung oder Formatänderung. Ist die Intensität in

[3] Ein LSB ist bei einer Binärzahl, hier der binären Darstellung eines Pixels, das Bit mit der niedrigsten Wertigkeit, also das Bit, das für 2^0 steht.

bestimmten Pixelblöcken stärker als in anderen oder kann es in Bereichen nicht mehr ausgelesen werden, so kann davon ausgegangen werden, dass das Bild unerlaubt manipuliert wurde.

Eine Kombination aus Medienmerkmalen und Wasserzeichen stellen die inhalts-fragilen Wasserzeichen dar. Hier werden inhaltsbeschreibende Merkmale robust als Wasserzeicheninformation eingebettet. Allerdings ist Kapazität[4] von robusten Wasserzeichen zurzeit noch nicht ausreichend um adäquate Merkmale vollständig in das beschriebene Medium einzubetten. Deswegen müssen Abstraktionen (z.B. Prüfsummen) von Merkmalen eingebettet werden, was die Sensibilität gegenüber Veränderungen herabsetzt oder die Lokalisierung von Veränderungen stark einschränkt [SD03].

1.3 Integration in Publikationsprozesse

Die bisher genannten Verfahren können zwar zuverlässig die Integrität digitaler Bilder belegen, sie erfordern aber die Integrationen zusätzlicher Schutzmechanismen in den Publikationsprozess. Auf der einen Seite erzeugen die geeigneten kryptografischen Schutzmechanismen zusätzliche, separate Daten (z.B. Hashes oder Merkmale), die zusammen mit dem Bildmaterial durch den Publikationsprozess gebracht werden müssen. Auf der anderen Seite ereichen die wasserzeichenbasierten Verfahren noch nicht die geeignete Empfindlichkeit gegenüber inhaltverändernden Manipulationen beziehungsweise ändert sich das, was als inhaltsverändernd angesehen wird, mit jedem Bild.

Wünschenswert wäre es zum einen also, keine separaten Zusatzinformationen mit durch den Publikationsprozess führen zu müssen und zum anderen, wenn Benutzer, die ihr Bild schützen möchten, für jedes Bild speziell angeben könnten, welche Operationen mit welcher Intensität an dem Bild vorgenommen werden dürfen, damit das Bild vertrauenswürdig bleibt. Für Benutzer, die das Bild bearbeiten möchten, sollte dann aber unmittelbar während der Bearbeitung ersichtlich werden, welche Änderung den Integritätsschutz verletzt hat. Dies fasst auch gleichzeitig die Anforderungen an den Bildeditor zusammen.

2 Entwurf eines wasserzeichentreuen Bildeditors

Als Lösung verfolgt diese Arbeit den Ansatz ein (fragiles) Wasserzeichen in ein Bild einzubetten, in dem Bildmerkmale wie Bildkanten und bestimmte Sicherheitseinstellungen gespeichert sind. Dieses Bild würde bei einer Bearbeitung in einem beliebigen Editor seinen Integritätsschutz verlieren. Verwendet man den hier vorgestellten wasserzeichentreuen Bildeditor, so werden die Sicherheitseinstellungen ausgelesen und nur Verarbeitungsschritte ermöglicht, die beim Einbettungsprozess des Wasserzeichens erlaubt wurden. Zusätzlich werden während der Bearbeitung die Bildmerkmale des bearbeiteten Bildes mit den Bildmerkmalen des Originalbildes, welche aus dem Wasserzeichen gelesen wurden, verglichen.

Beim Entwurf des wasserzeichentreuen Bildeditors soll die Benutzerfreundlichkeit eine wichtige Rolle spielen. Der Bildschutz sollte den Publikationsprozess nicht ausweiten, sondern sollte sich sinnvoll darin eingliedern (vgl. 1.3). Betrachtet man die Abläufe des Publikationsprozesses, so spielt der Bildschutz eine zentrale Rolle für Urheber bzw. Herausgeber eines Bildes, die den Integritätsschutz mit den individuellen Sicherheitseinstellungen einbetten

[4] Menge der einbettbaren Information

müssen. Für Bildbearbeiter, die das Originalbild für die Publikation nachbearbeiten möchten, spielt die Wahrung der Integrität, zu der sie verpflichtet sind, eine entscheidende Rolle, da gerade im Webbereich die Veröffentlichung mit starken Änderungen einhergehen kann.

Zunächst wird der Aufbau des wasserzeichentreuen Bildeditors beschrieben und anschließend an Hand der einzelnen Schritte des Publikationsprozesses erläutert, wie durch die signierte Einbettung von Merkmalsvektoren und Sicherheitseinstellungen mit Hilfe von Wasserzeichen, Bearbeitungsmöglichkeiten im Editor eingeschränkt werden können und damit die Integrität des Bildes im gesamten Publikationsprozess gewahrt werden kann.

2.1 Aufbau

Der wasserzeichentreue Bildeditor stellt eine Kombination aus digitalen Wasserzeichen und einem Digital Rights Management (DRM) System dar. Im gesamten Produktionsprozess werden ausschließlich Bilder in ihren gebräuchlichen Formaten (JPEG, GIF, etc.) verwendet. Sonderformate für das DRM oder angehängte Rechtedateien sind nicht notwendig. Durch digitale Wasserzeichen direkt in das Bild eingebrachten Informationen werden eingesetzt, um bestimmte Funktionalitäten des Editors freizuschalten bzw. einzuschränken.

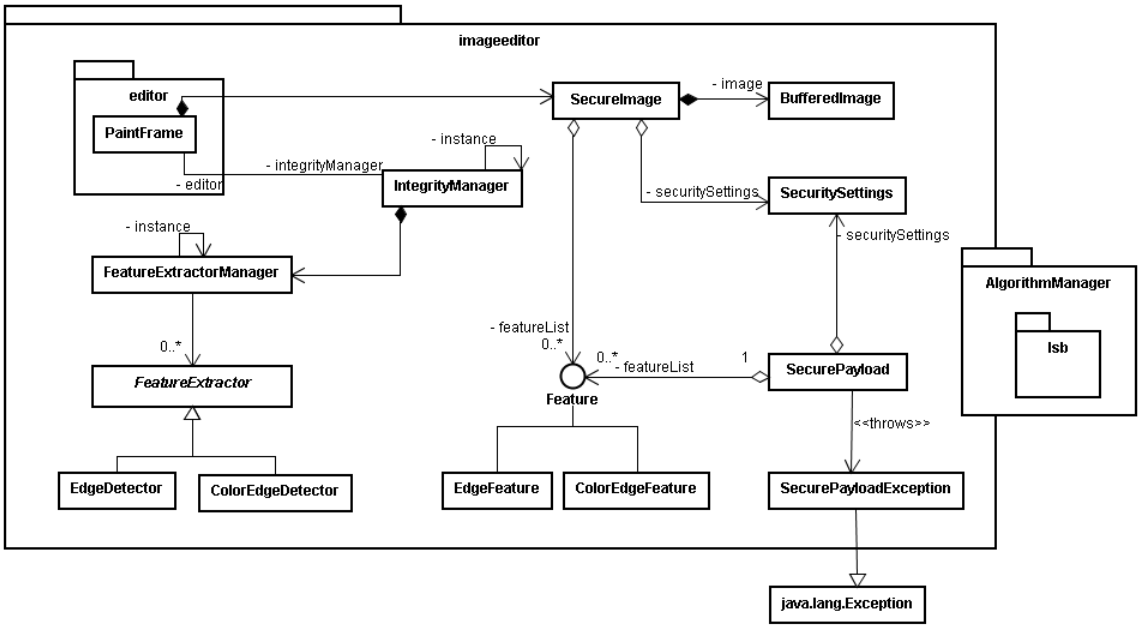

Abbildung 1: Klassendiagramm des wasserzeichentreuen Editors

Nach außen hin, präsentiert sich der Editor als handelsübliches (einfaches) Bildbearbeitungsprogramm, mit dem Bilder geladen, auf alle möglichen Weisen verändert und wieder abgespeichert werden können (s. Abbildung 1). Das für den Nutzer unsichtbare Kernstück des Editors bildet der `IntegrityManager`, der alle nachfolgend erläuterten Prozesse koordiniert.

Zunächst werden von einem geladenen Bild `Feature` ermittelt. Ein `Feature` repräsentiert einen Merkmalsvektor wie zum Beispiel Kantenverläufe oder Farbhistogramme wie sie in 1.2.1 erläutert wurden. Zu jedem `Feature` muss es einen passenden `FeatureExtractor` geben, der in der Lage ist, das entsprechende Merkmal aus dem Bild zu extrahieren. Welche

FeatureExtractor dem IntegrityManager zur Verfügung stehen, verwaltet ein FeatureExtractorManager. Dabei können für ein Bild auch mehrere Feature genutzt werden.

Ein im Editor geladenes Bild wird zum SecureImage, wenn ihm SecuritySettings hinzugefügt werden. Diese Sicherheitseinstellungen geben an, welche Editierfunktionen zukünftig auf diesem Bild erlaubt sein sollen und können beim Speichern eines Bildes definiert werden. So kann zum Bespiel angegeben werden, dass Formatkonvertierungen und Skalierungen um bis zu 20% erlaubt sein sollen, aber keine Ausschnittsbildung. Die nicht erlaubten Funktionen werden dann vom IntegrityManager für die weiter Benutzung gesperrt.

Wird durch den IntegrityManager das SecureImage mit einem Integritätsschutz versehen, so wird aus den Featuren und SecuritySettings sowie einigen Metadaten (wie etwa Größe etc.) eine SecurePayload erzeugt, welche digital signiert wird und durch ein Wasserzeichenverfahren aus dem AlgorithmManager in das Bild eingebettet wird. Beim Laden eines bereits geschützten Bildes wird das Wasserzeichen vom IntegrityManager ausgelesen. Damit stehen die ausgelesenen Merkmalsvektoren und Sicherheitseinstellungen des SecureImage zur Verfügung und können vom Editor berücksichtigt werden.

2.2 Arbeitsablauf

Am Anfang des Publikationsprozesses steht ein ungeschütztes Originalbild in den Händen der Urheber oder Herausgeber. Bei der Anwendung des wasserzeichentreuen Bildeditors gibt es zwei Ziele: das Schützen des Originalbildes auf der einen Seite und das Bearbeiten des geschützten Bildes unter Beachtung der Schutzeinstellungen, und damit automatisch auch der Wahrung der Integrität, auf der anderen Seite.

2.2.1 Schützen von Bildern

Zunächst können Urheber ihr Bild vor der Einbettung des Schutzes mit den Optionen des Bildeditors bearbeiten. Am Ende dieses Editiervorgangs steht das zu schützende Original, für das nun die Sicherheitseinstellungen festgelegt werden. Anschließend werden die Merkmalsvektoren des Originals (und die Metadaten, dir wir von nun an auch als Feature auffassen werden) extrahiert und mit den Sicherheitseinstellungen zur SecurePayload zusammengefasst. Diese wird anschließend noch von den Urhebern mit einer digitalen Signatur versehen, welche ebenfalls eingebettet wird.

Da die SecurePayload durch ein Wasserzeichenverfahren direkt in die dem SecureImage zu Grunde liegenden Bilddaten eingebettet wird, ist das Ergebnis des Erzeugungs- und Schutzprozesses – wie nach der Bearbeitung mit einem anderen Bildeditor auch - ein „ganz normales“ Bild, d.h. alle weiteren Schritte im Publikationsprozess laufen genau so ab, wie bei nicht geschützten Bildern auch. Damit ist eine nahtlose Integration in den Publikationsprozess erfolgt. Die Eigenschaften dieses Integritätsschutzes hängen wesentlich an konkreten verwendeten Wasserzeichenverfahren. Dieses wird in 2.3.2 näher untersucht.

2.2.2 Bearbeiten geschützter Bilder

Wird ein Bild im Editor zur Bearbeitung geöffnet, so wird automatisch versucht die SecurePayload auszulesen. Kann ein Integritätsschutzwasserzeichen ausgelesen werden, so ist versichert, dass das Bild integer ist.

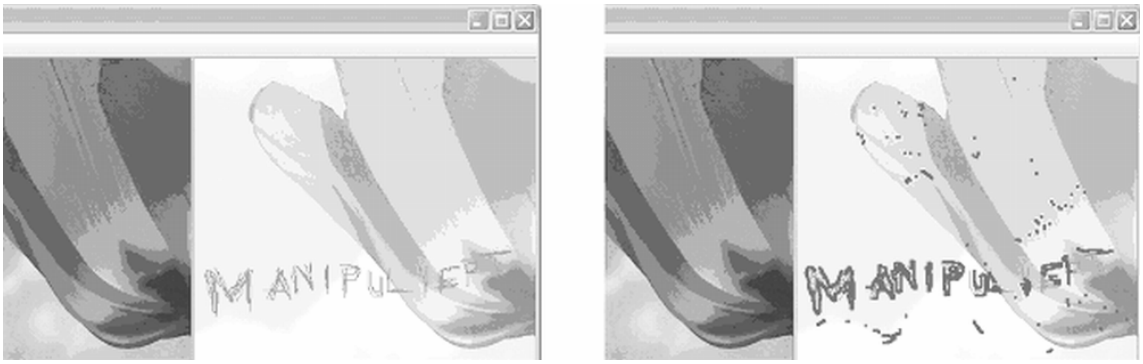

Abb. 2: Differenzanzeige im Bildeditor (links: deaktiviert, rechts: aktiviert)

Soll das Bild nun mit dem Integritätsschutz veröffentlicht werden, müssen bei der weiteren Bearbeitung die Sicherheitseinstellungen der Urheber oder Herausgeber berücksichtigt werden.

Die Merkmalsvektoren enthalten eine kompakte Beschreibung des ursprünglichen Bildes. Die Sicherheitseinstellungen bestimmen, in welchem Ausmaß Änderungen zugelassen werden. Damit können zum einen direkt bestimmte Funktionen des Editors deaktiviert werden, aber zum anderen auch nach jedem Editierschritt überprüft werden, ob dieser Schritt integritätserhaltend war. Dies geschieht, in dem vom bearbeiteten Bild die gleichen Merkmale extrahiert werden und – unter den aktuellen Sicherheitseinstellungen – mit den *eingebetteten* `Features` verglichen werden. Wird der Bearbeitungsschritt als integritätsverletzend erkannt, kann entweder der Schritt verhindert oder, benutzerfreundlicher, angezeigt werden, wo nicht erlaubte Veränderungen detektiert wurden und nur das Speichern des Bildes verhindert werden.

Waren alle Editierschritte integritätserhaltend, kann das Bild abgespeichert werden. Dabei wird die *ursprüngliche* `SecurePayload` wieder in das Bild eingebettet, so dass z.B. mehrfaches erlaubtes Skalieren um jeweils 10 % nicht in der Summe zu einer Integritätsverletzung führen kann.

2.3 Implementierung

In diesem Abschnitt erläutern wir kurz, welche konkreten Merkmale und welches Wasserzeichen für die Implementierung verwendet wurden und wie das Endergebnis aussieht.

2.3.1 Feature Detection

Der wasserzeichentreue Bildeditor ist so entworfen, dass er prinzipiell beliebige (auch mehrere) Merkmalsextraktionsalgorithmen verwenden kann. Für die erste Implementierung verwenden wir eine Javaimplementierung des Edison-Kantenextraktionsalgorithmus von Helfman [Hel03]. Dieser `EdgeDetector` und das dazugehörige `EdgeFeature` wurden erweitert zu einem `ColorEdgeDetector` (bzw. `ColorEdgeFeature`), in dem Kantenbilder von nur jeweils einer der drei Grundfarben als Grundlage für die Kantendetektion verwendet wurde.

Das `EdgeFeature` erlaubt einen pixel- bzw. blockweisen Vergleich zwischen Original und Bearbeitung und damit auch hinreichend genaue Lokalisierung der Integritätsverletzung, die direkt an Bearbeiter weitergegeben werden kann.

Weitere wichtige Informationen, die sich zum Schutz der Integrität als notwendig erwiesen haben, sind sog. Metadaten über das Bild. Hierzu zählen zum Beispiel Höhe, Breite und der verwendete Farbraum, die zu einem `MetaFeature` zusammengefasst und mit eingebettet werden können.

2.3.2 Wasserzeichenalgorithmus

Um die `SecurePayload` in das Bild einzubetten, muss der `AlgorithmManager` einen Wasserzeichenalgorithmus bereitstellen. Wie schon beim `FeatureDetector` zuvor, steht der Bildeditor prinzipiell jedem Wasserzeichenalgorithmus offen. Allerdings bestimmen die Eigenschaften des verwendeten Algorithmus wesentlich das Verhalten des Integritätsschutzes. In der vorliegenden Implementierung haben wir uns für ein fragiles Wasserzeichen mit hoher Kapazität entschieden. Deshalb müssen die Features nicht stark komprimiert werden und dadurch werden wiederum eine gute Lokalisierung von Manipulationen und eine hohe Genauigkeit bei der Erkennung von Manipulationen erreicht. Die Fragilität stellt sicher, dass ein außerhalb des Editors bearbeitetes Bild direkt nicht als integer erkannt wird. Hierzu verwenden wir ein LSB-Wasserzeichen, das die zur Einbettung verwendeten Bits je nach Zielformat des Bildes auswählt.

Eine weitere Möglichkeit zur Wahl des Algorithmus wäre, unterschiedliche Wasserzeichen für die Signatur der `SecurePayload` und die Payload selbst zu verwenden, um so auch nach externer Bearbeitung noch ursprüngliche Urheber ermitteln zu können. Für die Signatur sollte dann ein robustes Verfahren verwendet werden und für die `SecurePayload` z.B. ein Annotationswasserzeichen.

2.4 Diskussion

Der generische Aufbau des Editors ermöglicht es, verschiedenste Merkmale und Wasserzeichenverfahren einzusetzen. Die Sicherheit des Integritätsschutzes hängt aber natürlich stark an der konkreten Auswahl eingesetzter Merkmale und Verfahren.

Generell gilt, dass der Integritätsschutz folgende Grundkomponenten umfassen muss:

- Eine Auswahl bedeutungstragender Merkmale und deren Extraktion aus dem Original. Bedeutungstragend meint solche Merkmale, die sich bei inhaltsverändernder Bearbeitung des Bildes ebenfalls ändern. Im Idealfall sollten diese auch eine Lokalisierung der Integritätsverletzung erlauben.
- Sicherheitseinstellungen, die individuell für jedes Bild konkrete Beschreibungen enthalten, welche Operationen erlaubt sein sollen und welche nicht.
- Die Sicherheitseinstellungen und die Merkmalsvektoren müssen den Urhebern ein(ein)deutig zugeordnet sein.
- Bei Änderungen (zumindest bedeutungstragender Merkmale) außerhalb der sicheren Editorumgebung wird der Integritätsnachweis zerstört.

2.4.1 Aktuelle Implementierung

Die für die erste Implementierung verwendeten Kantenmerkmale erlauben die Entdeckung von lokalen Manipulationen, wie Entfernung oder Hinzufügen von Objekten oder Weichzeichnen. Dabei stellt die Verwendung dreier `ColorEdgeFeature` (Kanten werden separat

für jeden der drei RGB-Farbkanäle berechnet) sicher, dass einzelne Farbflächen nicht gegen andersfarbige Flächen gleicher Luminanz ausgetauscht werden können. Da Kanten aber keinen Schutz vor globalen Veränderungen (Filter), wie etwa Helligkeitsänderungen oder Formatkonversionen, bieten, können letztere direkt in den Sicherheitseinstellungen deaktiviert werden.

Die Fragilität des eingesetzten LSB-Wasserzeichens verhindert, dass das Bild unter Wahrung der Integrität außerhalb des Editors bearbeitet werden kann. Es ist möglich, die signierten LSB-Daten analog zu der Bearbeitung im Editor extern vor einer Veränderung zu speichern, die Änderungen durchzuführen und dann die LSB-Daten wieder in das Bild zu speichern (Copy-Attack). Hierdurch wird die Sicherheit allerdings nicht kompromittiert, da die Inhaltsmerkmale eventuelle Veränderungen beim Laden in den Editor anzeigen würden und diese nicht unbemerkt verändert werden können. Dieser Umstand gilt für alle Copy-Angriffe auf die eingesetzten Wasserzeichen.

Die Signatur der Urheber auf der `SecurePayload` ermöglicht zusätzlich den Nachweis der Authentizität. Gleichzeitig wird verhindert, dass andere Anwender die Bilder verändern und/oder neue Sicherheitseinstellungen vergeben und diese Manipulationen als auf die ursprünglichen Urheber zurückgehend darstellen.

2.4.2 Sicherheitsaspekte von Bildmerkmalen

Die Extraktion abgeleiteter Merkmale aus einer binären Bilddarstellung dient im Allgemeinen dazu, weg von einer Computer unterstützten Darstellung eines Bildes hin zu einer Repräsentation zu gelangen, die möglichst nahe an der menschlichen Wahrnehmung liegt. Somit kann eine bessere automatische Beurteilung erfolgen, ob eine Änderung am Bild Auswirkung auf die Merkmale und somit die Wahrnehmung desselben beim menschlichen Betrachter hat. Die Merkmalsextraktion dient folglich als Filter von für den Menschen relevanten und irrelevanten Bildinformationen.

Histogramme des gesamten Bildes können dazu dienen, Filterprozesse und andere globale Veränderungen aufzudecken. Dazu zählen Helligkeits- und Kontraständerungen oder Farbanpassungen (Stimmungsänderungen). Die Lokalisierung einer Verletzung ist zum Bespiel mit Hilfe von blockweise erstellten Histogrammen möglich – allerdings ist der benötigte Speicherbedarf linear von der Anzahl der Blöcke abhängig, was schnell zur Übersteigung der Kapazität des verwendeten Wasserzeichenalgorithmus.

Gewichtungen stellen eine hohe Abstraktion des Bildes dar, die gleichzeitig sensibel gegenüber Veränderungen an den Kanten ist. Durch einen vergleichsweise hohen Berechnungsaufwand wird hier eine sehr kompakte Inhaltsbeschreibung erkauft.

Fluchtpunkte verraten oft, ob Teile anderer Bilder in das zu schützende Bild eingefügt wurden oder ob das Bild gedreht wurde.

Ein grundsätzliches Problem bei der Überprüfung der Integrität durch mit einem Wasserzeichen eingebettete Bildmerkmale, ist die Tatsache, dass der Einbettungsprozess selbst auch eine Veränderung der Bilddaten darstellt. Dies kann je nach Bild und gewähltem Merkmal schon als integritätsverletzend angesehen werden. Die Bildmerkmale sind also eventuell nicht robust gegenüber der Einbettung von Wasserzeichen. Hier kann helfen, den Vergleich der eingebetteten und der extrahierten Merkmale nicht zu sensitiv zu gestalten, was mehr Spielraum für nicht aufgedeckte Integritätsverletzungen zulässt. Alternativ wären auch iterative

Ansätze denkbar, bei denen Konflikte zwischen Merkmalsextraktion und Wasserzeicheneinbettung durch wiederholtes Einbetten und Anpassen gelöst werden.

3 Zusammenfassung

Diese Arbeit hat mit dem wasserzeichentreuen Editor das Konzept einer Anwendung vorgestellt, welche die Möglichkeit bietet, Bilder mit einem Integritätsschutz zu versehen. Dieser Schutz bleibt auch dann noch gewährleistet, wenn die erstellten Bilder im weiteren Publikationsprozess noch bearbeitet werden müssen. Hierbei achtet der Bildeditor darauf, dass zwar Änderungen zur Bildoptimierung erlaubt sind, jedoch keine Änderungen am Bild vorgenommen werden dürfen, die die Bildaussage verändern. Der Integritätsschutz bleibt also während des gesamten Publikationsprozesses im wasserzeichentreuen Bildeditor erhalten, insofern der Bildeditor die Änderungen als inhaltserhaltend einstuft.

Der Bildeditor ist in der Lage, zwischen erlaubten Änderungen und Manipulationen, die die Bildaussage ändern zu unterscheiden, indem er Bildmerkmale des aktuellen Bildes mit denen des Originalbildes vergleicht. Die Bildmerkmale (Features) des Originals werden zum Schutz des Bildes mit einem Wasserzeichenverfahren in das Bild eingebettet und können jederzeit vom Editor ausgelesen werden. Da ähnliche Änderungen bei unterschiedlichen Bildern sowohl inhaltserhaltend als auch inhaltsverändernd sein können, erlaubt der wasserzeichentreue Editor Rechteinhabern beim Einbetten des Integritätsschutzes zu definieren, vor welchen Änderungen das Bild geschützt werden soll. So kann beeinflusst werden, ob etwa die Farben oder die Größe des Bildes auf jeden Fall erhalten werden müssen um den Integritätsschutz zu erhalten und damit weiterhin eine semantische Unversehrtheit des Bildes belegen zu können.

Danksagung

Die Arbeiten und Ergebnisse, die in dieser Veröffentlichung beschrieben sind, werden teilweise von der Europäischen Kommission innerhalb des IST Programms, Vertrag IST-2002-507932 ECRYPT, unterstützt.

Literatur

[B99] Buchmann; Einführung in die Kryptographie, Springer, Berlin, ISBN 3-540-66059-3, 1999

[CMB02] Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom: Digital Watermarking, Academic Press, San Diego 2002

[Dit00] Jana Dittmann: Digitale Wasserzeichen, Grundlagen, Verfahren, Anwendungsgebiete, Springer, Berlin 2000

[DS00] Jana Dittmann, Martin Steinebach: Manipulationserkennung bei digitalem Bildmaterial mit fragilen Wasserzeichen, in: Datenschutz und Datensicherheit, Verlag Vieweg, No. 10, 2000, S.593-597

[DSF+01] Jana Dittmann; Martin Steinebach; Lucialla Croce Ferri; Claus Vielhauer; Ralf Steinmetz; Petra Wohlmacher :Framework for media data and owner authentication based on cryptography, watermarking, and biometric authentication : Mul-

timedia Systems and Applications IV ;A. Tescher, B. Vasudev, M. Bove (Eds.) Proceedings of SPIE; , S. 198 - 209, SPIE, ISBN 0-8194-4242-9, 2001

[Fri93] Gary L. Friedman: The Trustworthy Digital Camera: Restoring Credibility to the Photographic Image, in: IEEE Transactions on Consumer Electronics, Vol. 39, No. 4, November 1993, S. 905-910

[Fri98] Jiri Fridrich: Image Watermarking for Tamper Detection, in: IEEE Interna-tional Conference on Image Processing, Vol. 2, Oktober 1998, S. 404-408

[Gol02] E. Bruce Goldstein: Wahrnehmungspsychologie, Hrsg. Manfred Ritter, 2. dt. Ausgabe, Spektrum Akademischer Verlag, Heidelberg 2002

[Hel03] Jonathan Helfman: *Java edge detection code*, "Online im Internet", http://www.fxpal.com/people/helfman/edge.html v. 2003, Abfrage v. 06.07.2005

[Jäh97] Bernd Jähne: Digitale Bildverarbeitung, 4. völlig neubearbeitete Auflage, Springer, Berlin 1997

[LFS04] Liu, Croce Ferri, Steinebach; Digital Watermarking for Integrity Protection of Synthetic Images, WIAMIS 2004; 5th International Workshop on Image Analysis for Multimedia Interactive Services, April 21-23, 2004, Instituto Superior Tonico, Lisboa, Portugal, ISBN 972-98115-7-1, 2004

[MG01] Peter Meer, Bogdan Georgescu: Edge Detection with Embedded Confi-dence, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No 12, Dezember 2001, S. 1351-1365

[SD03] Steinebach, Dittmann; Integrity Protection for digital audio, EURASIP Journal on Applied Signal Processing, Special Issue on Audio Processing, Volume No. 10, S. 1001 – 1015, Hindawi Publishing Corporation, 2003

[Ste04] Martin Steinebach: Digitale Wasserzeichen für Audiodaten, Dissertation, TU Darmstadt 2003

[Wal95] Steve Walton: Image Authentication for a Slippery New Age, in: Dr. Dobb's Journal, Vol. 20, No. 4, April 1995, S. 18-26

[WD96] Raymond B. Wolfgang, Edward J. Delp: A Watermark for Digital Images, in: IEEE International Conference on Image Processing, Vol. 3, September 1996, S. 219-222