FixBit-Container: Effizienter Urheberschutz durch Wasserzeichen entlang der Medien-Wertschöpfungskette

Patrick Wolf, Martin Steinebach

Fraunhofer SIT
Rheinstr. 75
64295 Darmstadt
patrick.wolf@sit.fraunhofer.de
martin.steinebach@sit.fraunhofer.de

Abstract: Beim Einsatz von Transaktionswasserzeichen zum Urheberschutz von digitalen Werken wird das zu schützende Werk mit einer Information versehen, die den Empfänger bzw. den Verteilvorgang (die Transaktion) kennzeichnet. Um eine lückenlose Sicherheit zu gewährleisten, müsste dies in jedem Schritt der Wertschöpfungskette geschehen, angefangen beim Rechteinhaber über Aggregatoren und Shops bis hin zum Endkunden. In der Praxis werden Wasserzeichen allerdings meist nur am Ende der Wertschöpfungskette eingesetzt, wohl um Mehrfachmarkierungen zu vermeiden. In dieser Arbeit stellen wir den FixBit-Container vor. Dieses für den praktischen Einsatz in Unternehmen entwickelte Prinzip erlaubt mit einem einzigen Markiervorgang und durch sukzessives Fixieren einzelner Bitpositionen die einzelnen Glieder der Medien-Wertschöpfungskette zu kennzeichnen. Dabei sind einzelnen Kennzeichnungsschritte nahezu so effizient wie ein Kopiervorgang.

1 Einleitung

In dieser Arbeit verfolgen wir einen aus der Praxis heraus motivierten, technisch-organisatorischen Lösungsansatz beim Einsatz von Transaktionswasserzeichen auf der gesamten Medien-Wertschöpfungskette – beginnend bei den Rechteinhabern bis hin zu Endkunden. Wissenschaftliche Erkenntnisse fließen mit Erkenntnissen aus der Unternehmenspraxis zusammen und münden im Konzept des FixBit-Containers, einer Mischung aus Vorgehensweise, Formatspezifikation und Arbeitsorganisationsstruktur.

1.1 Ausgangssituation

Transaktionswasserzeichen [SDN02] haben sich als Mittel zum Urheberschutz für alle Arten von Werken – angefangen bei Bild, Audio, Video und mittlerweile auch Ebooks – etabliert. Dabei wird eine Information über den Verbreitungsvorgang, etwa in Form einer Transaktionsnummer, in das zu schützende Werk eingebettet. Die Schutzwirkung beruht dabei nur zum Teil auf technischen Maßnahmen. Die Tatsache, dass durch die eingebet-

tete Information der Empfänger des Werks zu identifizieren ist, falls das Werk unberechtigterweise veröffentlicht und gefunden wurde [WS07], wirkt als psychologischer Schutz und verhindert so, dass der Empfänger seinerseits das Werk verbreitet. Bei der Vermarktung digitaler Werke kommen Transaktionswasserzeichen vor allem im letzten Glied der Wertschöpfungskette, beim Verkauf des Werks durch Online-Shops an Endkunden, zum Einsatz [SDN02]. Dabei haben die Online-Shops zunächst nur wenig originäres Interesse am Schutz der Werke, sondern werden durch Vertriebsverträge zum Schutz der Werke verpflichtet.

Die Rechteinhaber, die am Anfang der Wertschöpfungskette stehen, können aus naheliegenden praktischen Gründen, selbst wenn sie wollten, mit herkömmlichen Methoden keine mit Transaktionswasserzeichen individuell markierten Werke ausliefern. Sie müssten dann für jeden potenziellen Endkunden ein eigenes Werk bereitstellen. Damit könnten zwar analoge und digitale Auslieferung wieder völlig gleich behandelt werden, aber die Vorteile einer digitalen Auslieferung würden nicht zum Tragen kommen. Hinzu kommt, dass zum Urheberschutz durch Transaktionswasserzeichen die Markierung alleine nicht genügt – sie stellt nur einen passiven Vorgang dar [WSD07]. Als aktiver Teil dient die Suche nach unberechtigt veröffentlichen Werken im Internet. Die Rechteinhaber können natürlich die Online-Shops nicht nur zur Einbettung von Wasserzeichen verpflichten, sondern auch zu der Suche nach markierten Werken. Zur Suche gehört allerdings auch das Auslesen der Wasserzeichen und dieses hängt von den jeweiligen technischen Parametern des Markiervorgangs und insbesondere vom geheimen Schlüssel ab. Diese sind von Shop zu Shop wahrscheinlich unterschiedlich, so dass – selbst wenn man irgendwie die Suche und das Herunterladen von potenziell markierten Werken koordiniert bekommt – immer noch pro Shop ein aufwändiger separater Auslesevorgang durchgeführt werden muss. Es wäre wesentlich einfacher, wenn Rechteinhaber selbst Suchen nach markierten Werken beauftragen könnten.

Natürlich könnten Rechteinhaber die Werke, die sie an die nächsten Glieder der Medien-Wertschöpfungskette (z.B. Aggregatoren) ausliefern, mit Wasserzeichen versehen, um so den Vertriebsweg zu kennzeichnen. Auch im nächsten Schritt könnte das wieder geschehen bis hin zum Endkunden. Dafür wäre aber pro Glied ein Markiervorgang des selben Werks notwendig. Mehrfachmarkierungen sind zwar möglich [SZ08], aber die wahrgenomme Qualität eines markierten Mediums nimmt mit jedem Markierungvorgang höchstens ab.

Der Begriff Wertschöpfungs*kette* ist an dieser Stelle auch irreführend. Es wäre besser von einem Wertschöpfungs*baum* mit den Rechteinhabern als Wurzel zu sprechen, da auf jeder Ebene die Medien an mehrere Empfänger weitergegeben werden. Die Blätter dieses Wertschöpfungsbaums bilden dann die Konsumenten und jeder Pfad vom der Wurzel zu einem Blatt ist eine eigene Wertschöpfungskette. Um in jedem Knoten des Baums eine Markierung durchführen zu können, müsste jeweils eine vollständige Integration der Wasserzeichenalgorithmen in die jeweiligen Systeme durchgeführt werden, inklusive aufwändiger technischer Konfiguration. Erschwert wird dies noch zusätzlich dadurch, dass für Applikationen wie Online-Shops Java die vorherrschende Programmiersprache darstellt, für Wasserzeichensoftware aber meist C/C++ verwendet wird. Auch dies stellt kein prinzipielles Hindernis dar [WSZ08], erfordert aber vergleichsweise komplexe Schnittstellenarchitekturen.

1.2 Herausforderungen

Aus der Ausgangssituation lassen sich folgende konkrete, fachliche Herausforderungen herausarbeiten:

a) Die Rechteinhaber als Wurzel des Wertschöpfungsbaums werden in die Lage versetzt, jeden Weg zu einem Blatt mit Hilfe von Wasserzeichen nachzuvollziehen.

b) Rechteinhaber kontrollieren alleine den Einsatz von Wasserzeichen inklusive deren Auswirkungen auf die wahrgenommene Qualität ([CMB00]) des geschützten Werks. Insbesondere kennen sie alleine die notwendigen technischen Parameter und sind somit in der Lage auch nach mit Wasserzeichen markierten Werken zu suchen.

c) Die technischen Auswirkungen auf die Wertschöpfungskette sollen dabei minimal sein – insbesondere sollte das Handhaben der geschützten Werke möglichst in Java erfolgen.

d) Das eingesetzte Verfahren muss (trotzdem) hinreichend effizient sein, so dass ein Einsatz in Online-Shops technisch sinnvoll ist.

1.3 Struktur der Arbeit und Abgrenzung

Die oben genannten Herausforderungen werden in dieser Arbeit wie folgt angegangen: Zunächst wird in Kapitel 2 als Stand der Technik ein kurzer Abriss über den effizientem Einsatz von Wasserzeichen in Form der sogenannten Containertechnologie gegeben. Anschließend wird der existierende Containeransatz in Kapitel 3 erweitert, so dass er sich auf die gesamte Wertschöpfungskette ausdehnt. Dabei wird Wert auf tatsächliche praktische Erfahrungen im Unternehmenseinsatz gelegt und die Eigenschaften des FixBit-Containers beschrieben und analysiert. Ergebnis der Arbeit ist schließlich sowohl eine MPEG-ähnliche Formatdefinition wie auch ein Mechanismus zu deren Handhabung.

Es wird dabei bewusst nicht auf konkrete, die Container-Technologie implementierende Wasserzeichenverfahren oder deren (Sicherheits-)Eigenschaften eingegangen. Die Wasserzeichenverfahren werden als kryptologische Primitive angesehen und die Sicherheit des Gesamtsystems lässt sich voll auf diese Primitive zurückführen, da sich die durch den FixBit-Container markierten Medien nicht von herkömmlich markierten Medien unterscheiden lassen.

2 Stand der Technik: Containertechnologie

Das Konzept des digitalen Wasserzeichen-Containers stellt eine Lösung für die hohen Anforderungen an die Rechenleistung von Systemen beim Einbetten digitaler Wasserzeichen dar [SZC04, WHS08]. Wasserzeichenalgorithmen arbeiten in mehreren Schritten wie Signaltranformationen oder die Berechnung von psycho-perzeptiven Modellen. Die meisten

dieser Schritte sind unabhängig von der jeweils einzubettenden Nachricht. Gleichzeitig sind diese Schritte die rechenintesivsten. Die Idee des Wasserzeichencontainers ist es also, den Prozess des Einbettens von Wasserzeichen in zwei Phasen zu unterteilen: Eine Vorbereitungshase, die nur einmal durchgeführt wird und in der alle aufwändigen Berechnungen erfolgen und eine Markierungsphase, in der das individuell markierte Werk aus den vorberechneten Teilen zusammengesetzt wird. Abbildung 1 enthält die Phasen beiden (plus die Fixierungsphase aus Abschnitt 3).

2.1 Vorverarbeitungsphase

Für die Vorverarbeitungsphase wird zunächst die Länge der einzubettenden Nachricht festgelegt, genauso wie der geheime Wasserzeichen-Schlüssel und alle weiteren technischen Parameter.

Für den Container geeignete Wasserzeichennachrichten lassen sich als Folge individueller Bits („0“ und „1“) beschreiben und die jeweiligen zu markierenden Abschnitte des Mediums („*Elementary Units*“) müssen unabhängig voneinander markiert werden können. Dies führt dazu, dass für jede Elementary Unit nur eine Version mit einer „0“ und eine Version mit „1“ erstellt werden muss, um später beliebige Nachrichten zusammenstellen zu können.

Beide Versionen werden dann in einem eigens dafür definierten Containerformat abgespeichert. Das Erzeugen des Containers wird nur ein einziges Mal durchgeführt und ist in etwa so aufwändig wie das zweimalige Markieren des Werks.

2.2 Markierungsphase

Soll nun eine markierte Datei erzeugt werden, ist neben der Containerdatei nur noch die einzubettende Wasserzeichen-Information notwendig. Für jedes Bit der Nachricht wird jeweils die vormarkierte Version aus dem Container entnommen und so das individuell markierte Werk erzeugt. Da es sich bei diesem Vorgang im Wesentlichen um einen Lookup- und Kopiervorgang handelt, ist das Erzeugen eines markierten Werks sehr effizient. Zum Vergleich: Das individuelle Markieren einer Audiodatei, die im mp3-Format vorliegt, ist in der herkömmlichen Wasserzeichenvariante etwa viermal so schnell, wie die Abspieldauer des Musikstücks. In der Containervariante ist der gleiche Vorgang – nachdem der Container vorhanden ist – etwa 3000-mal so schnell wie die Abspieldauer [SZ08].

Die Markierungsphase kann räumlich und zeitlich getrennt von der Vorverarbeitungsphase ausgeführt werden. Insbesondere sind der geheime Schlüssel und die technischen Parameter in dieser Phase nicht mehr notwendig.

3 FixBit-Container

Die Container-Technologie erlaubt also bereits effizientes Markieren und sogar eine Trennung zwischen den wasserzeichentechnologisch komplexen Berechnungen und dem eigentlichen Erstellen des markierten Werks. Diese Grundeigenschaften werden für den FixBit-Container konsequent weiter ausgebaut. In diesem Kapitel wird aus Platzgründen eher auf das Prinzip des Containers und den daraus resultierenden Eigenschaften als auf das konkrete FixBit-Container-Format eingegangen.

3.1 Grundprinzip

Wie wir gesehen haben, kann Herausforderung a), die Nachvollziehbarkeit der genommenen Wege, befriedigt werden, wenn in jedem Verteilschritt der Wertschöpfungskette jeweils Transaktionswasserzeichen eingebracht werden. Diese Mehrfachmarkierung kann man aber auch als eine einzige Gesamtmarkierung auffassen, in dem die Wasserzeichennachrichten der Einzelmarkierung zu einer einzigen Markierung konkatiniert werden. Das Problem ist dann nur noch, dass die einzelnen Teile der Gesamtnachricht zu unterschiedlichen Zeitpunkten festgelegt werden. Für den FixBit-Container bedienen wir uns der Möglichkeit der räumlichen und zeitlichen Trennung, die uns die Container-Technologie bietet.

Die Grundidee ist, dass der FixBit-Container bereits am Anfang der Wertschöpfungskette erzeugt wird. Damit ist zunächst einmal automatisch Herausforderung b), die Forderung nach der Kontrolle der Wasserzeichenparameter durch die Rechteinhaber, erfüllt. Zusätzlich wird die Container-Technologie dahingehend erweitert, dass das Ergebnis der Markierungsphase nicht unbedingt ein vollständig markiertes Werk sein muss, sondern ein Container, bei dem nur ein Teil der Nachricht festgelegt wurde. So kann in jedem Schritt der Wertschöpfungskette zur Identifizierung des nächsten Glieds immer ein kleiner Teil der Nachricht fixiert werden bis schließlich beim letzten Auslieferungsschritt tatsächlich ein vollständig markiertes Werk entsteht und nicht mehr ein Container. Abbildung 1 erläutert dies im Detail.

Zur Erinnerung: Das zu markierende Werk besteht aus elementaren Einheiten („Elementary Units“ (EU) [WHS08]), die unabhängig voneinander jeweils mit einem einzelnen Bit markiert werden. Daraus entsteht wie bei der herkömmlichen Container-Technologie in der Vorbereitungsphase der Container. Aus einem solchen Container lässt sich jetzt in einer zusätzlichen Phase („Fixierungsphase“) ein Sub-Container, den eigentlichen FixBit-Container, erzeugen. Dies geschieht, in dem einzelne, aber nicht alle Bits fixiert werden und zwar einfach durch Löschung des jeweils komplementären Bits. Ergebnis ist ein Container, in dem nur noch Teile der Nachricht variabel sind – in der Abbildung sind die ersten beiden Bits auf jeweils „1“ fixiert worden. Aus einem FixBit-Container entstehen schließlich analog zur herkömmlichen Container-Technologie markierte Werke.

Das teilweise Festlegen einzelner Bits in der Fixierungsphase kommt einer Teilmarkierung gleich. Damit kann jedes Glied der Wertschöpfungskette nun das nächste Glied identifizieren, in dem es für jeden Empfänger andere Bits fixiert. Wird ein markiertes Werk gefunden

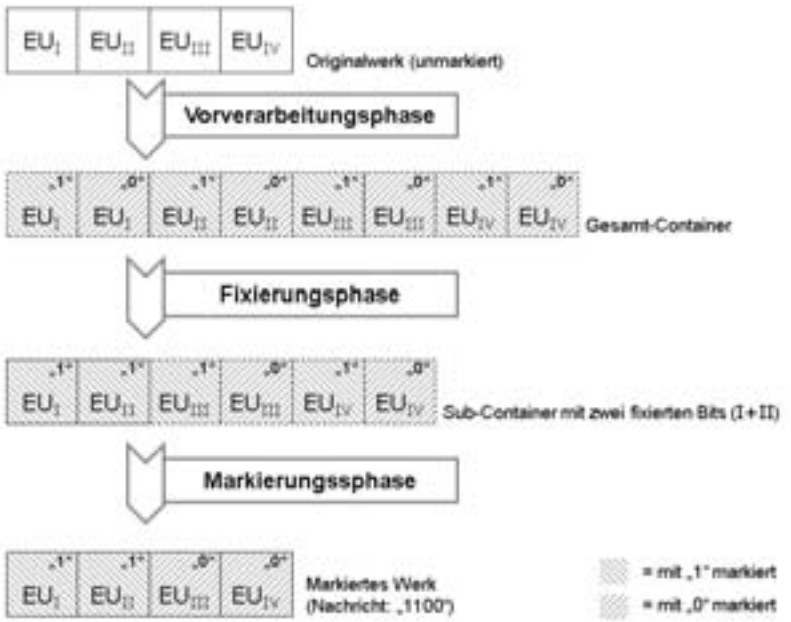

Abbildung 1: Phasen des FixBit-Containers

und die Nachricht ausgelesen, kann an Hand ihrer Zusammensetzung genau nachvollzogen werden, welchen Weg das Werk durch die Wertschöpfungskette genommen hat.

Ein Beispiel: Ein Rechteinhaber erstellt ein Container und legt fest, dass die Nachricht eine Länge von 48 Bit haben soll. Er will das Werk an einen Aggregator, seinen eigenen Online-Shop und an eine Promotion-Agentur verteilen. Um diese drei Empfänger auseinander zu halten, benötigt er nur 2 Bits der Nachricht. Er erzeugt aus dem Container drei FixBit-Container, wobei für den Aggregator die ersten beiden Bits der Nachricht auf „10", für seinen eigenen Shop auf „11" und für die Promotion-Agentur auf „01" fixiert. Dem Online-Shop bleiben damit noch 46 Bit der Nachricht übrig, um kundenindividuelle Nachrichten zu erzeugen.

Der Aggregator gibt das Werk nachfolgend an 20 Shops weiter. Dazu fixiert er (20-mal) fünf weitere Bits der Nachricht, so dass den Empfängern jeweils nur noch 41 Bit für ihre Nachrichten zur Verfügung stehen. Wird ein markiertes Werk gefunden, kann der Rechteinhaber an Hand der ersten beiden Bits feststellen, welchen Weg im Wertschöpfungsbaum das Werk genommen hat. Sind die ersten beiden Bit „10", so weiß er, dass er sich an den Aggregator wenden muss, um mit Hilfe der nächsten fünf Bit herauszufinden, an welchen Shop er sich wenden muss, um mit Hilfe der letzten 41 Bit die Identität des Empfängers zu erfahren.

3.2 Das FixBit-Container-Format und seine Eigenschaften

Das oben beschriebene FixBit-Container-Prinzip wurde umgesetzt und als Resultat entstand ein Container-Format, das ähnlich den MPEG-Standards zwar einen FixBit-Container beschreibt, sich aber nicht um dessen Erzeugung oder Interpretation kümmert. Durch die Standardisierung wird erreicht, dass jedes Container-Erzeugungsprogramm spezifisch und auf die Bedürfnisse des darunterliegenden Wasserzeichenalgorithmus angepasst ist, aber das Programm zum Erzeugen von individuell markierten Kopien aus Containern, der Shuffler, generisch sein kann – so wie MPEG-Dateien von unterschiedlichen Encodern erzeugt, aber alle vom selben Decoder abgespielt werden können. Insbesondere ist der Shuffler in Java implementiert und kann so sehr einfach in bestehende Businesssysteme integriert werden (für eine Geschwindigkeitsauswertung des Shufflers siehe Kapitel 3.3).

Das Format selbst ist ebenfalls an MPEG angelehnt und enthält zu den jeweiligen Untereinheiten des Werks (System Stream, Elementary Stream, Elementary Unit) Metainformationen in Form von Headern. Im Header einer Elementary Unit findet sich so zum Beispiel die Information, dass die nachfolgende Elementary Unit mit einer „1“ markiert wurde, 17554 Byte lang ist und ob sie direkt übernommen werden kann oder aus (Differenz-) Berechnungen aus der vorherigen Elementary Unit hervorgeht[1].

Durch diese Vorgehensweise ergeben sich für den FixBit-Container eine Reihe von für die Praxis wesentlichen Eigenschaften. Diese stellen auch gleichzeitig Antworten auf die Herausforderungen c) (minimale Auswirkungen) und d) (Effizienz) dar.

Volle Kontrolle über technische Parameter bei der Container-Erzeugung

Technische Parameter insbesondere der Wasserzeichenschlüssel werden ausschließlich bei der Erzeugung des ersten Containers benötigt. Dies versetzt den Erzeuger in die Lage, zum einen direkt die Qualität der markierten Werke sicherzustellen und gleichzeitig sind am Beginn der Wertschöpfungskette somit alle Information vorhanden, um nach mit dem Container erzeugten Werken suchen zu lassen.

Medientyp- und Medienformatunabhängigkeit

Das Container-Format ist sowohl vom Medientyp (Audio, Video, Bild, etc.) des zu schützenden Werks als auch von seinem Medienformat (mp3, PCM, FLAC etc.) unabhängig.

Möglichkeit zur Medienformat-spezifischen Nachbereitung

Trotz der Unabhängigkeit von Medientyp und –format kann es je nach Medienformat notwendig sein, das erzeugte Werk noch einmal nachbearbeiten zu müssen. Dies ist immer dann notwendig, wenn sich gewisse Eigenschaften des markierten Werks nicht direkt vorhersehen lassen bzw. sich eine Eigenschaft von zwei unterschiedlich markierten Werken

[1] Letzteres geschieht, wenn eine Elementary Unit nicht in Gänze im Container abgespeichert wird, sondern als Differenzsignal zum unmarkierten Original

unterscheidet (wie etwa die Dateigröße). Die Programme, die Sub-Container und markierte Werke erzeugen, müssen also trotz der Medienformatunabhängigkeit des Container-Formats in der Lage sein, Medienformat-spezifische Funktionalität auszuführen. Dies wird über „Plugins“ erreicht, die dynamisch beim Aufruf des Programms mitgeliefert werden können.

Streamingfähigkeit

In der Wertschöpfungskette gibt es zwei wesentliche Arten der Datenauslieferung: File-basiert und als Stream. Ersteres kommt beim Vertrieb von Einzelwerken in Shops zum Einsatz, letzteres ist vor allem im Broadcastbereich gebräuchlich. Das Container-Format ist so angelegt, dass es auch gestreamt werden kann.

Möglichkeiten der Verschlüsselung

Die einzelnen elementaren Einheiten können auch – wiederum unabhängig voneinander – verschlüsselt werden. Damit integriert das Container-Format auch die Eigenschaften des Krypto-Containers [SK07].

Byte-genauer Zugriff

Das FixBit-Container-Format ist so aufgebaut, dass Byte-genau auf das zu erstellende Werk zugegriffen werden kann. Dies spielt etwa beim Einsatz von Download-Managern in Online-Shops eine Rolle, wo mehrere Verbindungen zum markierten Werk geöffnet werden, das markierte Werk selbst sich aber nie auf eine Festplatte materialisieren soll. Erreicht wird diese Genauigkeit durch Angaben der tatsächlichen Größen jeder Elementary Unit.

3.3 Diskussion

Die Mehrheit der Herausforderungen (a)-c)) ist durch das Design des FixBit-Cotainers bereits erfüllt. Es bleiben allerdings einige Punkte offen, die es zu diskutieren gilt:

Geschwindigkeit

Offen bleibt etwa, ob die Java-Implementation des Shufflers tatsächlich den selben Effizienzgrad wie frühere, native Implementierungen des Container-Konzepts erreicht. Hierzu wurden Vergleichstests an Werken verschiedener Größe vorgenommen. Die Werke wurden auf einem herkömmlichen Desktop-PC jeweils 500 Mal sowohl mit der konventionellen als auch mit der Fix-Bit-Container-Technologie geshuffelt. Wie Abbildung 2 zeigt, werden mit den auf Java-basierten Shuffler die Werke ähnlich schnell wie mit dem nativen, konventionellen Shuffler erzeugt – bei kürzeren Stücken sogar schneller und meist mit

weniger Schwankungen der Shuffle-Zeiten. Dabei liegen die Datenraten nur etwa 10% bis 15% unterhalb der Datenschreibrate der Festplatte.

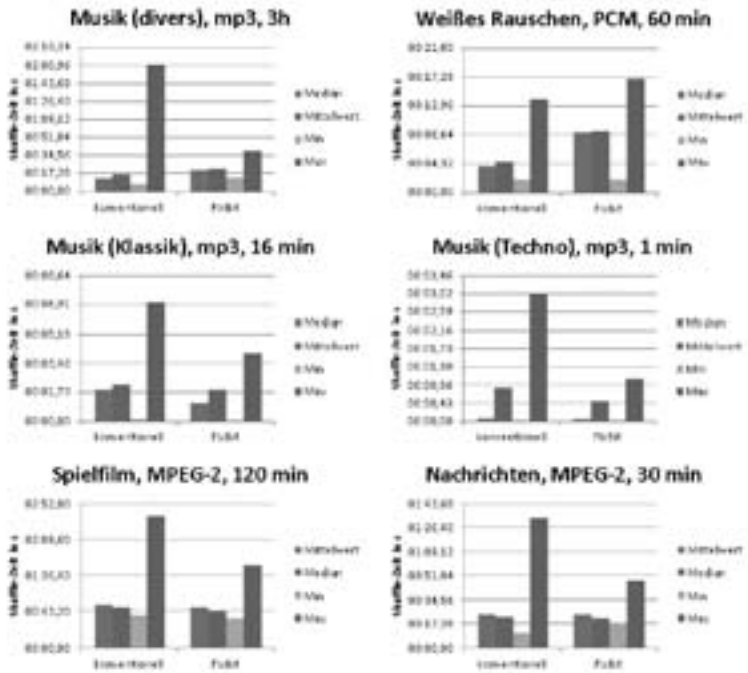

Abbildung 2: Messung Shuffle-Zeiten diverser Werke bei 500-fachem Shuffling

Prüfsummen und Fehlerkorrektur

Wasserzeichennachrichten werden im Allgemeinen durch Prüfsummen und Fehlerkorrektur geschützt [Ber08]. Um während des letztens Shufflens eine Prüfsumme über die Gesamtnachricht berechnen zu können, ist es notwendig, dass der Shuffler diese vollständig kennt. Sonst müsste jede Teilnachricht mit eigenen Prüfsummen versehen werden, was die Kapazität deutlich reduziert. Dazu enthält das Container-Format Angaben, welche Bits mit welchem Wert fixiert wurden, so dass auch für einen FixBit-Container Prüfsummen genutzt werden können.

Bei Fehlerkorrektur-Verfahren wird dagegen die „Netto“-Nachricht zunächst in eine fehlerkorrigierender Darstellung („Brutto“) verwandelt und diese eingebettet. Zusätzlich werden in der Brutto-Darstellung mehrere Netto-Bits miteinander verknüpft. Dies führt dazu, dass in einem FixBit-Container nicht beliebige Teilnachrichten in Fehler-korrigierender Darstellung fixiert werden können. Anderseits ist es möglich [Ber08] durch das Hinzufügen einzelner weniger Brutto-Bits einen Zustand zu erreichen, in dem ein Teil einer

Netto-Nachricht unabhängig von ihren anderen Teilen wird. Dieses Verfahren ist auch auf den FixBit-Container übertragbar.

Koalitionssicherheit und Fingerprinting

Man spricht von einem Koalitionsangriff gegen ein Wasserzeichen, wenn aus Werken mit unterschiedlichen Markierungen ein neues Werk zusammengesetzt wird. Um solchen Angriffen zu begegnen werden sog. Fingerprinting-Codes [SBH+10] eingesetzt. Diese Codes sind wesentlich länger als herkömmliche Wasserzeichennachrichten und werden so erzeugt, dass aus beliebigen Kombinationen der unterschiedlichen Werke immer mindestens eine Version, die an der Angriffskoalition beteiligt war, sicher identifiziert werden kann.

Beim Einsatz von Fingerprinting-Codes werden nicht einfach beliebige Nachrichten festgelegt, sondern die einzelnen Codes (vorab) erzeugt. Insofern lassen sich nicht einfach FixBit-Container und Fingerprinting-Code mischen. Allerdings ist es möglich, die Menge der Fingerprinting-Codes in Untergruppen übereinstimmender Teile einzuteilen. Auf diese Weise kann man sicherstellen, dass alle Codes, die an ein Glied der Wertschöpfungskette weitergereicht werden, gemeinsame gleiche Passagen enthalten. Diese können dann fixiert werden und agieren so als identifizierende Teilnachrichten.

4 Zusammenfassung und nächste Schritte

In dieser Arbeit haben wir eine praktische Lösung präsentiert, wie Rechteinhaber bereits an der Wurzel des Medien-Wertschöpfungsbaums Wasserzeichen zum Einsatz bringen können. Der FixBit-Container ist sowohl eine Vorgehensweise wie auch ein standardisiertes Weitergabeformat für Wasserzeichen, das erlaubt, zur Identifizierung jeder Abzweigung des Wertschöpfungsbaums einen Teil der Wasserzeichennachricht als Teilnachricht zu fixieren. Trotzdem bleibt die Fähigkeit erhalten, an den Blättern des Baums effizient markierte Werke zu erhalten.

Rechteinhaber haben nun auf diese Weise sowohl Kontrolle über die Qualität der Markierung ihrer Werke und können gleichzeitig die Suche nach ihnen unabhängig vom konkreten Auslieferungsweg in die Hand nehmen.

Ein spannender nächster Schritt wird die Entwicklung von Container-Verfahren für nichtkontinuierliche Medien wie etwa Bilder sein, bei denen sich das zu markierende Medium nicht einfach in Elementary Units zerlegen lässt wie bei kontinuierlichen Medientypen wie Audio und Video.

Danksagung

Diese Arbeit wurde unterstützt durch CASED (`http://www.cased.de/`).

Literatur

[Ber08] Waldemar Berchthold. Optimierung der Robustheit und Klangqualität digitaler Audio-Wasserzeichen-Verfahren im Kontext von angepassten Vorwärtsfehlerkorrektur-Algorithmen. Diplomarbeit, Hochschule Darmstadt, Fachbereich Mathematik, 2008.

[CMB00] I. J. Cox, M. L. Miller und J. A. Bloom. Watermarking applications and their properties. In *Proc. Int Information Technology: Coding and Computing Conf*, Seiten 6–10, 2000.

[SBH+10] Marcel Schäfer, Waldemar Berchtold, Margareta Heilmann, Sascha Zmudzinski, Martin Steinebach und Stefan Katzenbeisser. Collusion Secure Fingerprint Watermarking for Real World Applications. In *Proceedings of GI Sicherheit 2010*, 2010.

[SDN02] Martin Steinebach, Jana Dittmann und Christian Neubauer. Anforderungen an digitale Transaktionswasserzeichen für den Einsatz im e-Commerce. In Klaus P. Jantke, Wolfgang S. Wittig und Jörg Herrmann, Hrsg., *Von e-Learning bis e-Payment - Das Internet als sicherer Marktplatz*, Seiten 209 – 217, Leipzig, September 2002. Akademische Verlagsgesellschaft Aka GmbH.

[SK07] Martin Steinebach und M. Kaliszan. Kryptographisch geschützte Wasserzeichencontainer. In Patrick Horster, Hrsg., *DACH Security 2007*, Seiten 370–380, Klagenfurt, 2007. syssec verlag.

[SZ08] Martin Steinebach und Sascha Zmudzinski. Evaluation of robustness and transparency of multiple audio watermark embedding. In E. Delp, P. Wong, J. Dittmann und N. Memon, Hrsg., *Security, Steganography, and Watermarking of Multimedia Contents X*, Bellingham, 2008. SPIE and IS&T.

[SZC04] Martin Steinebach, Sascha Zmudzinski und Fan Chen. The digital watermarking container: secure and efficient embedding. In *Proceedings of the 2004 workshop on Multimedia and security*, MM&Sec '04, Seiten 199–205, New York, NY, USA, 2004. ACM.

[WHS08] Patrick Wolf, Enrico Hauer und Martin Steinebach. The video watermarking container: efficient realtime transaction watermarking. In E. Delp, P. Wong, J. Dittmann und N. Memon, Hrsg., *Security, Steganography, and Watermarking of Multimedia Contents X*, Bellingham, 2008. SPIE and IS&T.

[WS07] Patrick Wolf und Martin Steinebach. *Multimedia Forensics and Security*, Kapitel On the Necessity of Finding Content Before Watermark Retrieval: Active Search Strategies for Localising Watermarked Media on the Internet. IG Global, 2007.

[WSD07] Patrick Wolf, Martin Steinebach und Konstantin Diener. Complementing DRM with digital watermarking: mark, search, retrieve. *Online Information Review*, 31(1):10–21, 2007.

[WSZ08] Patrick Wolf, Martin Steinebach und Sascha Zmudzinski. Adaptive security for virtual goods - building an access layer for digital watermarking. In Rüdiger Grimm, Hrsg., *Virtual Goods. 6th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods incorp. the 4th International ODRL Workshop*, Poznan, Poland, October 2008.