Digital Rights Management

Matthias-Christian Ott

9. November 2010

Inhaltsverzeichnis

1 Begriffsklärung

Der Begriff DRM ist eine Abkürzung für „Digital Rights Management“. Allgemein bezeichnet der Begriff Technologien, die ermöglichen zu kontrollieren, wie digitale Inhalte benutzt und verbreitet werden können.

Die Bedeutung des Begriffs „digitale Inhalte“ ist allgemein schwer zu definieren. Praktisch sind zur Zeit damit Software, Texte, Audio- und Videoinhalte, die einen marktwirtschaftlichen Wert haben und zugleich auf digitalen Geräten oder Medien gespeichert sind und wiedergegeben werden, gemeint.

DRM-Systeme bestimmen, wann, wo und von wem auf digitale Inhalte zugegriffen werden darf und von wem, wie oft und mit welchen Rechten die Inhalte vervielfältigt und verbreitet werden dürfen. Diese Nutzungsregeln werden aber nicht vom Benutzer, sondern vom Rechteinhaber, der die immateriellen Rechte an den jeweiligen Inhalten hält, bestimmt und durchgesetzt. Der Benutzer stimmt dabei dieser Fremdbestimmung vertraglich zu.

2 Technische Funktionsweise

Ein DRM-System hat mehrere technische Ebenen: Es speichert digitale Inhalte und schützt diese gegen unbefugte Zugriffe und Vervielfältigungen. Zur Umsetzung muss in einer maschinenlesbaren Beschreibungssprache festgelegt sein, welche Rechte dem Benutzer gewährt und verwehrt werden. Das umfasst auch den Widerruf von Rechten aus der Ferne. Einige DRM-Systeme enthalten auch die Möglichkeit Inhalte, sofern sie durch Umgehung des DRM-Systems widerrechtlich verbreitet werden, bis zum Ursprung zurück zu verfolgen.

2.1 Datenspeicherung

Damit ein DRM-System obigen Anforderungen genügt und wirksam ist, dürfen also die digitalen Inhalte, die über dieses verwaltet werden sollen, nur innerhalb und nicht ohne das System zugänglich und lesbar sein.

Das Datenformat, in dem die Inhalte vorliegen, darf also nicht öffentlich bekannt sein. Es muss also entweder verschleiert (obfuscated) und geheim sein oder die Daten müssen verschlüsselt und nur vom DRM-System selber entschlüsselbar sein.

Geheimhaltung und Verschleierung

Damit Außenstehende den Daten des DRM-Systems keine Bedeutung zuordnen und damit nicht lesen können, halten sehr viele DRM-Systeme ihre Datenformate geheim oder verschleiern diese[1] [9].

Oft werden auch schwache kryptografische Algorithmen, die das Kerckhoffsche Prinzip [27, S. 12] missachten, eine unzureichende Schlüssellänge haben, veraltet und angreifbar oder proprietär sind und keinem Peer-Review oder einer Zertifizierung unterzogen wurden, als Verschleierung und nicht als Verschlüsselung eingeordnet, da sie verglichen mit den Industriestandards nur unzureichende Sicherheit bieten.

Verschlüsselung

Verschlüsselung dient dazu, eine vom Sender verschlüsselte Nachricht nur für einen Empfänger, der den entsprechenden kryptographischen Schlüssel, sofern die Chiffre schlüsselbasiert ist, oder

[1] Die einzige mir bekannte Ausnahme ist das von Sun Microsystems, Inc. entwickelte DReaM DRM, das unter einer Freien Software Lizenz steht und dessen Wirksamkeit auf Trusted Computing beruht.

die Chiffre kennt, lesbar zu machen.

Damit Verschlüsselung in DRM-System funktioniert, darf also nur der Rechteinhaber und das System selber, aber nicht der Benutzer des Systems, den Schlüssel haben oder die Chiffre[2] kennen. Das DRM-System muss also den Schlüssel oder die Chiffre vor dem Benutzer geheim halten.

2.2 Confinement

DRM-Systeme versuchen das Confinement Problem [29] zu lösen. Das Confiment Problem besteht darin zu verhindern, dass ein Programm A Daten zur Verarbeitung an ein Programm B mit festgelegten Rechten weitergibt, gleichzeitig aber verhindern möchte, dass B diese Daten und damit verbundenen Rechte an ein Programm C weitergibt. Die Daten von A sollen also in B eingesperrt werden (confined).

Wenigstens auf Betriebssystemebene lösen einige auf Capabilities basierende Betriebssysteme dieses Problem, indem das Betriebssystem als überwachende Autorität dafür sorgt, dass bestimmte Sicherheitsprinzipien eingehalten werden [32].

DRM-Systeme versuchen in diesem Sinne auch als überwachende Autorität aufzutreten, um sicherzustellen, dass der Benutzer Inhalte und Rechte nur nach den von dem jeweiligen Rechteinhaber festgelegten Regeln weitergibt. Der wesentliche Unterschied besteht jedoch darin, dass weder das DRM-System noch die dazugehörigen Regeln vom Benutzer kontrollierbar oder veränderbar sein dürfen. Könnte der Benutzer sie kontrollieren, so könnte er das DRM-System anweisen oder es so verändern, dass es die Regeln ignoriert oder teilweise nicht oder anders anwendet, oder die Regeln abändern.

2.3 White-Box Cryptography

Im Gegensatz zum üblichen Black-Box-Modell, in dem ein Angreifer nur die Ein- und Ausgabe einer Chiffre auslesen kann und daraus den kryptografischen Schlüssel, geht White-Box-Cryptography davon aus, dass ein Angreifer vollständige Kontrolle über die Ausführungsumgebung hat, das umfasst die Kontrolle über die Programmausführung, Hauptspeicherinhalte und Festspeicher [56, S. 70 ff.]. Ziel von White-Box-Cryptography ist es, durch Verschleierung den Schlüssel im Algorithmus zu „verstecken“, so dass zwar die Chiffre uneingeschränkt genutzt, aber der Schlüssel nicht extrahiert werden kann. Auf diese Weise soll verhindert werden, dass für einige DRM-Systeme wichtige Hauptschlüssel bekannt werden[3].

2.4 Digitale Wasserzeichen

Digitale Wasserzeichen dienen in DRM-Systemen zumeist der Kennzeichnung der Urheberschaft. Ein digitales Wasserzeichen kann sichtbar oder unsichtbar sein. Sichtbare Wasserzeichen sind,

[2] Im Folgenden ist dies als inklusives „oder“ zu verstehen, da sowohl Schlüssel als auch Algorithmus geheim sein könnten.

[3] Es ist mir dabei nicht wirklich klar geworden, wie dies das DRM-System schützen soll, da der Angreifer trotzdem in Besitz von Schlüssel und Chiffre ist, nur den Schlüssel nicht auslesen kann. Auf Grund des Umfangs konnte ich [56] und verwandte Veröffentlichungen aber nicht vollständig lesen, so dass ich mir über diesen Widerspruch kein abschließendes Urteil bilden konnte. Es scheint aber absurd, dass der Angreifer auf der einen Seite in seiner Rolle als Benutzer die Chiffre zum Entschlüsseln von Inhalten nutzen soll und auf der anderen Seite volle Kontrolle über das System und die Chiffre und Schlüssel hat, aber trotzdem nicht jegliche Inhalte damit entschlüsseln können soll. Auch die Verwendung von White-Box-Cryptography in einem Chiffre basierten Message Authentication Code (MAC) kann ausgeschlossen werden, da die Authentizität einer Nachricht entweder ignoriert werden kann (Angreifer ist Empfänger der Nachricht) oder der MAC-Algorithmus auch ohne Kenntnis des Schlüssels gebraucht werden kann (Angreifer ist Sender der Nachricht). Dadurch, dass aber eine spezifische Implementierung der Chiffre benutzt wird, wird das Einbetten von für Traitor Tracing nützlichen Informationen möglich und unumgehbar, sofern der Schlüssel trotz aller Verschleierung nicht doch extrahiert werden und damit eine andere Implementierung der Chiffre benutzt werden kann.

sehr ähnlich zu analogen Wasserzeichen, eine wahrnehmbare, bei der Betrachtung nicht großartig störende, aber nicht zu entfernende Kennzeichnung. Ein unsichtbares oder steganografisches digitales Wasserzeichen ist eine nicht sinnlich wahrnehmbare in den digitalen Inhalt eingebettete Information, zum Beispiel ein nicht wahrnehmbares Rauschen in einem Musikstück. Ein ideales Wasserzeichen lässt sich auch durch analoge Ausgaben (zum Beispiel Abfilmen oder Ausdrucken) oder hohen Qualitätsverlust schwer bis gar nicht entfernen.

2.5 Traitor Tracing

Einige DRM-Systeme enthalten die Möglichkeit, Inhalte, die unrechtmäßig außerhalb des DRM-Systems verbreitet werden – das umfasst auch insbesondere analoge Kopien, die beispielsweise durch Abfilmen, Mitschneiden oder das Aufzeichnen analoger Ausgabesignale gemacht werden – zurück zu verfolgen, um die Person, von welcher der Inhalt anfänglich verbreitet wurde, zu ermitteln und diese rechtlich zu verfolgen.

In einem Traitor Tracing Schema[4] [12, 52] enthält jeder digitale Inhalt zu diesem Zweck einen für einen Benutzer spezifischen digitalen Fingerabdruck. Verbreitet ein Benutzer einen Inhalt weiter, so lässt sich dieser bis zu dem Benutzer zurück verfolgen, dessen Fingerabdruck sich in dem Inhalt befindet.

2.6 Hardware

Die Funktionsweise eines DRM-Systems muss also entweder geheim sein, so dass eine Veränderung durch den Benutzer schwer möglich ist, oder eine Veränderung des DRM-Systems muss durch die Geräte, auf denen es aufgeführt wird, auf Hardware-Ebene verhindert werden. Zur Zeit sind die meisten DRM-Systeme proprietäre Software und haben den ersten Weg gewählt. Durch Technologien wie Trusted Computing ist aber auch der zweite Weg möglich und wird teilweise schon eingesetzt.

Trusted Computing

Ein komplette technische Beschreibung von Trusted Computing würde den Umfang dieser Arbeit übersteigen[5], deshalb beschränkt sich die folgende Beschreibung vor allem auf nicht-technische, oberflächliche Beschreibung der Anwendung von Trusted Computing auf Digital Rights Management.

Im Allgemeinen stellt das für Trusted Computing notwendige Trusted Platform Modul (TPM) kryptografische Funktionen und Speicher zur Verfügung. Dadurch, dass das Modul in Hardware implementiert ist, soll es für den Benutzer möglich sein, sich auf die Sicherheit dessen verlassen zu können. Anders aufgefasst muss er dem Modul allerdings auch vertrauen, denn das Modul ist eine Blackbox mit definierter Schnittstelle und wurde auch mit dieser Zielsetzung entwickelt, da ein Teil der Sicherheit auf dieser Tatsache basiert. Auf diese Weise kommt dem Wort „trusted“ eine zu bedenkende Doppelbedeutung zu.

Ein für DRM sehr wichtiges Anwendungsgebiet von Trusted Computing ist Trusted Boot. Mit Hilfe des TPMs wird sämtliche Software vom Start des ersten Programmcodes über den Bootloader und das Betriebssystem und Treiber bis zu den eigentlichen Anwendungen verifiziert und entschlüsselt. So kann verhindert werden, dass die zum DRM-System gehörenden Programme ausgelesen oder manipuliert werden. Des Weiteren können digitale Inhalte auch mit Hilfe des TPMs

[4] Es existieren verschiedene Traitor Tracing Schemata in der Literatur, die ich weder beurteilen noch überblicken kann. Ich beschränke mich hier auf eine einfache Variante, wie sie zum Beispiel auch in Advanced Access Content System (AACS) verwendet wird.

[5] Alleine die Spezifikation Trusted Platform Modules (TPM) umfasst 696 Seiten und diese beschreibt im Wesentlichen nur die Funktionen, die das TPM implementieren muss. Es existieren aber noch etliche weitere Spezifikationen der Trusted Computing Group, die zum Beispiel Schnittstellen zu anderen Technologien beschreiben.

entschlüsselt und zugehörige Schlüssel im TPM gespeichert werden, so dass diese Funktionen nicht von dem DRM-System in Software implementiert werden müssen. Vorausgesetzt, dass das TPM nicht manipulierbar ist, wird so ein Angriff auf das DRM-System unmöglich.

3 Letting Loose the Light

Der 1994 verfasste und 1996 erschiene Essay „Letting Loose the Light“ [45] des bei Xerox PARC arbeitenden Wissenschaftlers Mark Stefik stellt die auch aus heutiger Sicht noch unverwirklichte Idee eines technisch und gesellschaftlich umfassenden DRM-Systems vor.

Der Text wurde in der Entstehungsphase des modernen DRMs geschrieben und hatte großen Einfluss auf die Entwicklung und Vorstellung von Digital Rights Management [38, S. xii].

„Letting Loose the Light“ hebt sich dadurch ab, dass nicht nur eine technische, sondern auch eine umfassende gesellschaftliche Zukunftsvorstellung präsentiert wird, die sich nicht nur auf die Sicht einzelner Inhalteanbieter beschränkt, sondern DRM als einen gesellschaftlichen Prozess auffasst, der nach Auffassung Stefiks zum Wohle aller stattfinden müsse und historisch unabdingbar sei, wenn man Kreativität im digitalen Zeitalter erhalten wolle.

Stefik wendet sich gleichzeitig aber auch gegen die sich mit dem aufstrebenden Internet verbreitende, durch Steward Brand mit dem einflussreichen Ausspruch „information wants to be free“[6] beschriebene Vorstellung der unkontrollierten und unregulierten und damit letztendlich auch unentgeltlichen Verbreitung von Informationen und Inhalten[7], die durch Computernetzwerke möglich geworden war, denn Stefiks Auffassung nach führe die freie und unkontrollierte Verbreitung von Informationen und Inhalten zu Chaos und der Zerstörung jeglicher Kreativität durch Versagen des Marktes.

Stefiks Position und die Rolle von DRM allgemein ist in diesem weit über DRM hinausgehenden gesellschaftlichen Konflikt [33] klar auf der Seite der Kontrolle und Regulation, die zwar auf vertraglicher Basis freiwillig geschieht, aber durch die breite gesellschaftliche Teilnahme, die aus der Übernahme der bestehenden Verhältnisse entsteht, vorherrschendes Modell der Informationsverbreitung sein soll.

3.1 Trusted System

Grundlage von Stefiks DRM-System bilden vertrauenswürdige oder verlässliche Systeme (Trusted Systems).

Entgegen ihrem Namen unterschieden sich solche Systeme von Allzweckcomputern, mit denen der Benutzer alles machen könne, was mit der Hardware möglich sei, darin, dass sie dem Benutzer stets misstrauen würden. Sich auf ein Trusted System verlassen oder ihm vertrauen sollen nur die Entwickler können, denn jedes Trusted System solle vor Manipulation geschützt sein. Damit ein Trusted System nicht manipulierbar sei, müsse folgendes gewährleistet sein:

- Ein Trusted System müsse gegen jegliche physikalische Manipulationen geschützt sein (physical integrity), darunter fielen: Öffnen des Gehäuses, Austausch und Ausbau von Komponenten, Manipulation von Komponenten. Im Falle einer Manipulation müssten sich die Daten, die in dem System gespeichert sind, auch selbst zerstören können.
- Ein Trusted System dürfe nur mit anderen Trusted Systems kommunizieren und Daten austauschen (communicational integrity). Auf diese Weise wird das Confinement Problem gelöst.

[6] „On the one hand information wants to be expensive, because it's so valuable. The right information in the right place just changes your life. On the other hand, information wants to be free, because the cost of getting it out is getting lower and lower all the time. So you have these two fighting against each other.“ ([5, S. 49])

[7] „[…] changing the confrontational issue of fee versus free to a practical issue of ‚how much?'“ ([45, S. 30])

Durch Verwendung asymmetrischer Kryptografie und Zertifikatssperrlisten sei es möglich, eine gegenseitige Authentizität sicher zu stellen und manipulierte Repositories durch Widerrufen ihres Zertifikats von jeglicher Kommunikation auszuschließen. Es bedürfe dafür einer zentralen Zertifizierungsstelle, die Stefik Master Repository nennt.

- Ein Trusted System müsse stets sämtliche Funktionen gemäß der Anforderungen ausführen (behavioral integritry).

3.2 Repository

Damit die digitalen Inhalte geschützt seien, würden sie ausschließlich in Repositories gespeichert und reproduziert (abgespielt, angezeigt, ausgeführt). Damit dieses wirksam sei, müssten Repositories Trusted Systems sein, denn nur so sei sichergestellt, dass die dem Benutzer erteilten Nutzungsrechte stets eingehalten werden.

Der Begriff „Repository“ umfasse unter anderem Speichermedien, eingebettete Systeme, Bildschirme, Drucker und klassische Computer. Das DRM-System müsse dabei möglichst umfassend sein und dürfe nur aus Trusted Systems bestehen.

Repositories mit analoger Ausgabe würden eine besondere Herausforderung darstellen, da die Ausgabe auf der einen Seite für die menschliche Wahrnehmung nötig sei, aber auf der anderen Seite analoge Inhalte uneingeschränkt verbreitet werden könnten (analogue gap), da diese das Trusted System ungeschützt verlassen würden. Dies könne zu einem gewissen Maße mit digitalen Wasserzeichen und Steganographie verhindert werden, indem beispielsweise ein Scanner verweigern würde, ein Dokument, das von einem Trusted Printer – einem Drucker, der gleichzeitig ein Trusted System ist – mit einem Wasserzeichen versehen wurde, einzuscannen [46]. Des Weiteren sei der Schaden, der durch unrechtmäßige analoge Kopien entstehe, begrenzt, da mit einer analogen Kopie ein Qualitätsverlust verbunden sei.

Credit Server

Damit sichergestellt sei, dass alle in Anspruch genommenen Rechte bezahlt würden, der Bezahlvorgang automatisch geschehe und einfach für den Benutzer sei, müsse ein Repository ein Bezahlsystem eingebaut haben.

Ein solches System, das von Stefik Credit Server genannt wird, müsste auch nicht ständig mit einem Kreditinstitut verbunden sein, so dass es auch für mobile oder eingebettete Systeme verwendet werden könnte. Der Benutzer würde in diesem Fall einen Kreditrahmen eingeräumt bekommen, nach Überschreiten dessen der Credit Server sich dann wieder mit dem Kreditinstitut synchronisieren müsste. So sei die Schadenshöhe, die durch Zerstörung des Gerätes ohne Synchronisation enstehen könnte, begrenzt. Eine Versicherung könnte das Restrisiko abfedern.

3.3 Usage Rights

Jeder in dem DRM-System gespeicherte Inhalt müsse mit Nutzungrechten (usage rights) verbunden sein, damit ein Repository wisse, welche Rechte dem Benutzer für einen bestimmten digitalen Inhalt zu gewähren seien.

Grundsätzlich seien zwei Arten von Nutzungsrechten der Reproduktion von Inhalten zu unterscheiden: Play Rights und Print Rights.

Damit Inhalte von einem Repository A in ein anderes Repository B übertragen werden könnten, wie dies zum Beispiel beim Verkauf eines Inhalts geschieht, müsse es möglich sein, Rechte für diesen Vorgang, so genannte Transfer Rights, einzuräumen. Diese würden dann zum Beispiel auch sicherstellen, dass im Falle eines Verkaufs entweder der Inhalt in A gelöscht oder der Credit Server im Repository B mit dem Kaufpreis des digitalen Inhalts belastet würde.

Der Verleih von Inhalten ließe sich durch Verleihrechte (loan rights) abbilden. Würde ein Inhalt verliehen, so würde der im Repository des Verleihenden deaktiviert und in das Repository des Ausleihenden übertragen. Nach Überschreitung der Leihfrist oder Rückgabe würde der Inhalt dann automatisch wieder aus dem Repository des Ausleihenden gelöscht.

Digital Rights Language

Um Mehrdeutigkeiten, die mit natürlichen Sprachen verbunden sind, zu vermeiden, müssten die Nutzungsrechte in einer formalen Sprache mit definierter formaler Syntax und Semantik festgehalten sein. Diese müsse standardisiert und von allen Repositories gleich interpretiert werden. Es müsse also Interoperabilität geben. Gleichzeitig müsse sich bestehendes Recht in ihr ausdrücken lassen und die Sprache müsse an sich verändernde Verhältnisse anpassbar sein.

3.4 Digital Property Trust

Um das DRM-System langlebig zu machen, müsse es nachhaltig und auch durch zukünftige Generationen nutzbar sein. Es müsse eine Organisation, die Stefik den Digital Property Trust nennt, die dies zum Ziel habe, geben.

Ein wichtiger Bestandteil dieser Nachhaltigkeit sei die Standardisierung und die damit verbundene Schaffung von Interoperabilität. Dies umfasse die Standardisierung der Digital Rights Language, um sicherzustellen, dass gleiche Rechtsvorstellungen zwischen den verschiedenen Herstellern herrschten, die Schaffung von Interoperabilität zwischen Trusted Systems verschiedener Hersteller – dies geschehe durch die Zertifizierung von Trusted Systems – und das Betreiben des Master Repository.

Die andere wichtige Aufgabe des Digital Property Trusts sei ein sozialer Ausgleich zwischen den verschiedenen Interessengruppen[8]. Wie Stefik in einer späteren Arbeit ausführte [46], solle der Trust aus Vertretern der Verlags- und Medienwirtschaft, Herstellern von Trusted Systems und von Finanzinstitutionen, Politikern und Vertretern der Bibliothekare und Konsumenten bestehen. Nur so ließe sich gegenseitiges Vertrauen und ein gesellschaftlicher Kompromiss ähnlich des Copyrights schaffen.

Der soziale Ausgleich sei gleichermaßen als eine Art soziale Verantwortung zu sehen, die darin bestehe, spezielle Lizenzen für Arme, Wissenschaftler und Universitäten, Lehrer und Schulen und Bibliotheken zur Verfügung zu stellen. Solche Lizenzen seien natürlich nur an vertrauenswürdige Personen auszustellen, denen man vertrauen könnte, ihre Privilegien nicht zu missbrauchen, und die dies gegenüber entsprechenden Institutionen unter Beweis gestellt hätten.

Die gleiche Verantwortung sei aber auch gegenüber der Verlags- und Medienindustrie aufzubringen, da es immer eine gewisse Anzahl von Rechtsverletzungen geben würde, denn keine Technologie sei vollkommen. Stefik schlägt zur Entschädigung die Einrichtung einer Versicherung vor, in die zu einem geringen Anteil mit der Transaktion einbezahlt würde.

4 Beispiele

Es ist schwierig, gerade bei einer relativ jungen Technologie wie DRM in der heutigen Zeit verlässliche Prognosen oder Aussagen über Technologien und deren Wechselwirkung mit der Gesellschaft zu machen.

Dennoch kann man anhand verbreiteter DRM-Systeme deren Lebenszyklen und Wechselwirkungen betrachten und rückblickend auf ein Jahrzehnt DRM die frühe Zukunftsvision Stefiks mit den

[8] Stefik fasst die Situation wie folgt zusammen: „Trusted systems do not exist in a vacuum. They exist in a social framework.“ ([46])

tatsächlich auf den Markt gebrachten Technologien vergleichen und versuchen, eine Entwicklungsrichtung dieser Technologie auszumachen, auch wenn diese vielleicht nicht vollständig repräsentativ ist.

4.1 Content Scramble System

Das Content Scramble System (CSS) ist ein 1996 von einem Industriegremium[9] beschlossenes Verschlüsselungssystem für Videoinhalte auf DVDs [3]. Es wurde entwickelt, um das Kopieren von DVDs zu unterbinden.

Durch die hohe Verbreitung von DVDs ist CSS eines der am weitesten verbreiteten DRM-Systeme und eines der wenigen frühen DRM-Systeme, welches noch heute eingesetzt wird. Des Weiteren ist CSS im Gegensatz zu vielen heutigen DRM-Systemen standardisiert, auch wenn der Standard nur nach Unterzeichnung einer Vertraulichkeitserklärung einsehbar ist.

Funktionsweise

CSS basiert auf einer proprietären Stromchiffre[10] mit 40-Bit Schlüssellänge. Die Schlüssellänge, die 1996 nicht ausreichend war, wurde jedoch auf Grund von Exportbeschränkungen in den USA auf 40-Bits begrenzt [14, S. 8] und war keine Entwicklungsentscheidung.

Die Daten einer DVD werden im CSS durch drei Typen von Schlüsseln verschlüsselt: den Geräteschlüssel, den Hauptschlüssel und den Titelschlüssel. Zusätzlich authentifizieren sich Laufwerk und Abspielgerät [17].

Jeder Hersteller von Abspielgeräten besitzt mindestens einen ihm von der DVD CA zugewiesenen Geräteschlüssel und speichert diesen bei der Produktion (entweder in Software oder in Hardware) in dem Gerät ab. Es teilen sich also alle Geräte eines Herstellers oder einer Produktlinie denselben Geräteschlüssel.

Der für jede DVD spezifische Hauptschlüssel ist zusammen mit einer Prüfsumme des Schlüssels auf der DVD jeweils mit allen bis zur Produktion der DVD bekannten Geräteschlüsseln verschlüsselt gespeichert.

Jeder auf der DVD vorhandene Titel ist mit einem eigenen Titelschlüssel verschlüsselt, der wiederum durch den Hauptschlüssel verschlüsselt ist.

Um die Daten einer DVD auslesen und sie damit abspielen zu können, muss ein Abspielgerät also zuerst (sofern vorhanden) den mit dem Geräteschlüssel verschlüsselten Hauptschlüssel entschlüsseln, diesen gegen die Prüfsumme prüfen, dann den Titelschlüssel des jeweiligen Titels entschlüsseln und schließlich mit dem Titelschlüssel die Daten entschlüsseln.

Verlauf

Dass es sich bei CSS um ein proprietäres Verschlüsselungssystem handelt, dessen Wirksamkeit als DRM-System von der Geheimhaltung der Geräteschlüssel und des Algorithmus und damit auch der Kontrolle der Verbreitung dieser Schlüssel abhängt, ist (zumindest in den USA) eine legale Nutzung in Freier Software nicht möglich, da Freie Software ohne Beschränkung verändert und verbreitet werden kann.

Das mit dem Aufkommen von DVD-Laufwerken und TV-Karten für den Heimbereich 1999 gegründete Linux Video and DVD Project (LiVid) hatte sich jedoch unter anderem zum Ziel gesetzt, DVDs auf dem freien Betriebssystem GNU/Linux abspielen zu können. Nach Auffassung einiger Entwickler bedeutete das auch mit CSS verschlüsselte DVDs abspielen zu können.

[9] Später ging dieses Gremium in die DVD Copy Control Association (DVD CCA) auf [3].
[10] Siehe [51] für eine detailierte Kryptoanalyse.

Im September 1999 wurde CSS gebrochen und die proprietäre Software DoD Speed Ripper veröffentlicht [24]. Daraufhin kündigte Jon Lech Johansen an, ein Freund arbeite an einer Freien Software, die er demnächst veröffentlichen würde [24, 26]. Nachdem Derek Fawcus angab, ihm sei der Quelltext einer funktionierenden CSS-Enschlüsselungssoftware zugespielt worden [24, 19], veröffentlichte Johannsen die Software DeCSS, die allerdings nur unter Microsoft Windows funktionierte.

Obwohl Jon Lech Johannsen wahrscheinlich nur die grafische Benutzeroberfläche für DeCSS geschrieben hatte, wurde er in Norwegen angeklagt, 2004 jedoch freigesprochen[11].

Die Anklage gegen Johannsen und Abmahnungen gegen Betreiber von Internetseiten, auf denen CSS-Entschlüsselungssoftware verfügbar war, führten zu einer großen Medienaufmerksamkeit und zur hohen Verbreitung der Software auf vielen Internetseiten.

Der von DeCSS verwendete Geräteschlüssel der Xing DVD-Abspielsoftware wurde nach dem Bekanntwerden zurück gerufen und nicht weiter auf DVDs verwendet.

Bewertung

Motivation, CSS zu umgehen, war Interoperabilität herzustellen und das Abspielen jeglicher DVDs auf GNU/Linux zu ermöglichen, also nicht wie später von Filmindustrie behauptet, um der Filmindustrie zu schaden und Filme unrechtmäßig zu kopieren. Andererseits hätte eine auf GNU/Linux funktionierende proprietäre Abspielsoftware den Verlauf auch nicht beeinflusst, denn Ziel der Entwickler war nicht einfach nur DVDs abspielen zu können, dafür hätten Sie auch ein externes Abspielgerät oder Microsoft Windows kaufen können, sondern Freie Software zu haben, die dies tut. Diese Forderung konnte natürlich von DVD CCA nicht erfüllt werden, da, wenn CSS als Quelltext vorgelegen hätte, es vollständig unwirksam gewesen wäre. In gewisser Weise war also Überzeugung vielmehr das Motiv CSS zu knacken als sich persönlich zu bereichern.

CSS basiert zwar auf einer schwachen Chiffre, doch diese wurde anfänglich gar nicht angegriffen und ist auch nicht der Grund, aus dem CSS geknackt wurde, denn die Funktionsweise von CSS wurde durch Reverse-Engineering bekannt. Selbst die beste damals bekannte Chiffre hätte also nicht verhindern können, dass CSS umgangen werden konnte, da dem Angreifer Chiffre und Schlüssel gleichzeitig ausgeliefert werden, was die Verschlüsselung unwirksam macht. Das Auslesen von Chiffre und Schlüssel hätte nur durch ein von Stefik vorgeschlagenes Trusted System, das damals wie heute nicht existiert, verhindert werden können.

Die auf die Veröffentlichung von DeCSS folgenden Versuche, dessen Verbreitung und die Verbreitung anderer CSS-Entschlüsselungssoftware zu verhindern, sind bis heute einmalig und zeugen davon, dass scheinbar nicht damit gerechnet wurde, dass CSS geknackt wird. Verhindert haben Gerichtsprozess und Abmahnungen nichts. Sie haben vielmehr dazu beigetragen, dass sich eine von einflussreichen Institutionen wie der FSF und EFF gestützte große Gegenbewegung gebildet hat, der durch die mediale Aufregung eine Möglichkeit eingeräumt wurde, ihre Ansichten zu verbreiten. Insofern haben sich die Inhalteanbieter dadurch noch weiter geschadet und im Endeffekt das Gegenteil ihrer Bestrebungen erreicht.

4.2 FairPlay

FairPlay ist ein dynamisches DRM-System der Firma Apple, Inc. für Musikstücke, Filme und E-Books. Es wurde am 28. April 2003 zusammen mit dem iTunes Music Store, der damals eine Verkaufplattform für Musikstücke für Apples Abspielgerät iPod war[12], eingeführt.

Das DRM-System sollte verhindern, dass Musikstücke, die im iTunes Music Store gekauft worden waren, illegal kopiert oder auf Abspielgeräte anderer Hersteller aufgespielt werden konnten.

[11] Der Freispruch wurde kontrovers aufgefasst, da Johannsen angab, er habe DeCSS mitentwickelt, um DVDs unter GNU/Linux abspielen zu können, DeCSS jedoch nur unter Microsoft Windows funktionierte.

[12] Nach Einführung anderer Inhalte wurde der iTunes Music Store in iTunes Store umbenannt.

FairPlay wurde nach Einführung des Verkaufs und Verleihs von Filmen auch auf Videos, bei Eröffnung des App Stores, einem Softwaremarktplatz für iOS[13] Anwendungen, auf iOS Anwendung und nach Erscheinen des von Apple verkauften Tablet-Computers iPad und der zugehörigen Lesesoftware iBooks auch auf E-Books ausgeweitet.

Funktionsweise

FairPlay assoziiert jeden gekauften Inhalt mit einem Benutzerkonto. Der Benutzer kann dann dem Benutzerkonto bis zu fünf Computer zuordnen, auf die gekaufte Inhalte überspielt und von diesen abgespielt werden können [15, 35].

Jeder digitale Inhalt innerhalb von FairPlay ist mittels AES mit einem Hauptschlüssel verschlüsselt. Jedes Benutzerkonto hat einen eigenen kryptografischen Schlüssel, der auf den Servern von Apple und lokal auf dem Computer des Benutzers in iTunes gespeichert ist. Wenn der Benutzer einen Inhalt kauft, so wird der Inhalt auf den Servern von Apple mit dem Hauptschlüssel verschlüsselt, der Hautpschlüssel in dem Inhalt gespeichert und der Inhalt auf den Computer des Benutzers übertragen, wo der Hauptschlüssel dann wiederum mit dem Benutzerschlüssel verschlüsselt wird. Zum Entschlüsseln des Inhalts sind also Haupt- und Benutzerschlüssel notwendig. Ein Computer mit iTunes kann dabei für mehrere Benutzerkonten autorisiert sein und damit mehrere Benutzerschlüssel haben.

Kopiert der Benutzer den Inhalt auf ein Gerät[14] von Apple, so wird auch der Benutzerschlüssel mitkopiert, damit mit diesem der Hauptschlüssel und mit dem Hauptschlüssel dann der Inhalt gelesen werden kann.

Da die Benutzerkonten auch an Kredikartendaten geknüpft sind, ist auch die Wahrscheinlichkeit, dass sich mehrere Personen ein Konto teilen, gering.

Verlauf

FairPlay wurde durch die von Jon Lech Johansen geschriebene Software QTFairUse, die die entschlüsselten Daten einfach aus dem Hauptspeicher ausliest, im Oktober 2003 erstmals geknackt [4]. Im April 2004 konnte die Software dann den Inhalt und den zugehörigen Schlüssel aus einem iPod auslesen [20].

Wenig später entstanden einige Programme, die in der Lage waren, Inhalte aus dem iTunes Store zu entschlüsseln. Anstatt die Inhalte erst mittels iTunes herunterzuladen und dann entschlüsseln zu lassen oder die Schlüssel auszulesen, emulierte beispielsweise PyMusique[15], eine von Jon Lech Johansen mitentwickelte Software, iTunes und lud die Inhalte ohne sie zu verschlüsseln von den iTunes Servern herunter.

Apple ging rechtlich gegen Anbieter und Entwickler solcher Software vor und versuchte die Löschung derer Internetseiten durchzusetzen [6, 13]. Des Weiteren versuchte Apple durch Updates von iTunes und Änderung des Protokolls andere Software auszuschließen, was aber nicht gelang [8]. Später wurde FairPlay dann auch für Filme umgangen.

In dem am 6. Februar 2007 erschienenen Essay „Thoughts on Music“ [25] versuchte Steve Jobs, Chairman und CEO von Apple darzulegen, dass DRM eine von der Musikindustrie gesetzte Voraussetzung für den Verkauf von Musik sei, auf die Apple keinen Einfluss hätte und deshalb selber dem Wunsch der Kunden nach DRM-freier Musik nicht nachkommen könnte. Des Weiteren sei eine Interoperabilität der iPods dadurch gegeben, dass auch Musikstücke ohne FairPlay, was 97 % der zur Zeit gespeicherten Daten seien, abspielbar seien. Eine Lizensierung von FairPlay zur Erhöhung

[13] iOS ist das zur Zeit auf Apples iPod und iPad verwendete Betriebssystem.

[14] Die Anzahl der Geräte ist dabei nicht begrenzt. Jedoch wird, anders als bei Abspielgeräten nahezu aller anderer Hersteller, der komplette Inhalt des Geräts mit iTunes synchronisiert, so dass, wenn ein Benutzer sein Gerät mit dem iTunes eines anderen Benutzers synchronisiert, zuerst alle Daten auf dem Gerät gelöscht werden.

[15] Später wurde die Software in C♯ neu geschrieben und in SharpMusique umbenannt.

der Interoperabilität an andere Firma sei aus Sicherheitsgründen auch nicht möglich.

Am 2. April 2007, nachdem Ende 2006 bekannt geworden war, dass Amazon, Inc. eine Verkaufsplattform für DRM-freie MP3s plane [22], gab Apple bekannt, alle Musikstücke von EMI würden gegen einen Aufpreis von 30 Cent ohne FairPlay angeboten, bereits gekaufte Musikstücke ließen sich für den reinen Aufpreis umwandeln [31]. Am 6. Januar 2009 gab Apple dann bekannt, alle verbleibenden noch nicht DRM-freien Musikstücke würden zu den gleichen Konditionen bis Ende März 2009 ohne DRM angeboten.

Filme, Bücher und iOS-Anwendungen sind bis jetzt jedoch nur mit FairPlay verfügbar [36].

Bewertung

Um FairPlay zu umgehen war es noch nicht einmal nötig, irgendeine Form von Kryptografie zu umgehen, da der Hauptschlüssel in den digitalen Inhalt eingebettet ist und erst nachdem der Inhalt übertragen wurde, von iTunes verschlüsselt wird. Wenn alternative Software lediglich das von iTunes und dem iTunes Store verwendete Protokoll implementiert, aber die Inhalte nicht nachträglich verschlüsselt, kann man noch nicht einmal davon sprechen, dass das DRM-System geknackt wird. Auf der anderen Seite hätte eine zusätzliche Ebene kryptografischer Verschleierung auch nichts daran geändert, dass weder die mit FairPlay ausgestatteten Apple Geräte noch iTunes ein Trusted System und damit manipular sind.

Erst nachdem Amazon mit Amazon MP3 eine DRM-freie Alternative zum iTunes Store angekündigt hatte und vor allem nachdem Apple genug Geräte verkauft hatte, um gut am Markt platziert zu sein und einen ausreichenden Lock-In-Effekt hergestellt zu haben, war Apple bereit, über DRM-freie Musikstücke für den iTunes Store zu verhandeln und versuchte dann auch noch sein Handeln medienwirksam positiv darzustellen.

Im Vergleich zu anderen ähnlich wirkungslosen DRM-System scheint es trotzdem erstaunlich, dass sich sowohl Geräte als auch mit FairPlay versehene digitale Inhalte gut verkauft haben.

4.3 Amazon Kindle

Der Amazon Kindle ist ein E-Book-Reader der Firma Amazon.com, Inc. mit einem E-Ink-Display.

Die erste Geräteversion ist im November 2007 erschienen und war 5 Monate lang nach dem Verkaufsstart ausverkauft [54]. Bis Dezember 2009 wurden schätzungsweise 3 Millionen Geräte verkauft [2]. Später ist auch Software für iOS, Mac OS X, Microsoft Windows und Android erschienen.

Funktionsweise

Der Amazon Kindle basiert auf einem DRM-System mit GNU/Linux als Betriebssystem, auf dem eine proprietäre in Java geschriebene grafische Oberfläche läuft [41, 43], die sich durch Kindlets mit dem Kindle Developer Kit erweitern lässt [28].

Das Gerät ist mit dem Onlineshop von Amazon verbunden, lädt von dort über das Mobilfunknetz gekaufte Inhalte (zumeist Bücher) herunter und speichert sie auf dem Gerät.

Das Gerät zeigt dabei nur verschlüsselte AZW- und Topaz-Dateien und unverschlüsselte MOBI- und Textdateien an und spielt MP3-, AAC- und WAV-Dateien sowie Hörbücher im Audible Format ab. JPEG-, GIF-, PNG-, BMP-, DOC-, RTF- und PDF-Dateien werden von Amazon in das AZW-Format konvertiert und entgeltlich auf den Kindle synchronisiert.

Das Mobipocket-Format ist ein proprietäres Datenformat zur Speicherung und Darstellung von E-Books. Es wurde im Jahr 2000 von der von Amazon 2005 aufgekauften Firma Mobipocket S.A. für den Vertrieb von E-Books auf ihrer Verkaufsplattform Mobipocket.com mit Hinblick auf damalige Mobilgeräte (PDAs, Mobiltelefone) entwickelt. Das Format beinhaltet auch eine primitive Unterstützung für DRM.

Grundlage des AZW-Formats ist das Mobipocket-Format. Der wesentliche Unterschied besteht in der Verwendung eines besseren Kompressionsalgorithmus und eine dadurch erreichte höhere Kompressionsrate. Insbesondere ist auch das DRM-System nahezu unverändert geblieben.

Das Topaz-Format ist ein proprietäres Dateiformat, das für den Kindle entwickelt wurde. Es unterscheidet sich wesentlich darin von AZW, dass eigene Glyphen eingebettet werden können. Das DRM-System ähnelt AZW.

Der sonst sehr weit verbreitete offene Formatstandard EPUB wird vom Amazon Kindle nicht unterstüzt. Vermutlich wollte Amazon damit konkurrierende Buchhändler, die dieselben Bücher im EPUB-Format teilweise günstiger anbieten, durch ausschließliche Verwendung eines eigenen proprietären Formats ausschließen.

Jeder von Amazon verkaufte Kindle hat eine Amazon bekannte eindeutige Gerätenummer [34]. Im Falle von AZW wird aus der Gerätenummer ein Personal Identifier (PID) erzeugt, der mit dem Hauptschlüssel mit der von Alexander Pukall entwickelten Stromchiffre PC1[16] verschlüsselt, das Ergebnis ist ein temporärer Schlüssel. Beim Kauf eines Inhalts wird dieser mit dem temporären Schlüssel verschlüsselt. Zur Anzeige des Inhalts generiert der Kindle auf die gleiche Weise den temporären Schlüssel und entschlüsselt damit den Inhalt.

Der PID zur Entschlüsselung des Topaz-Formats setzt sich zumindest bei Kindle4PC aus einer zufälligen Zahl, der Gerätenummer der Festplatte, dem Benutzernames des Computers, einem für das Amazon Benutzerkonto spezifischen Token und dateispezifischen Schlüsseln zusammen, wobei diese Informationen verschleiert und schließlich eine SHA1 Summe daraus berechnet wird. Mit der PID wird dann der eigentliche Schlüssel entschlüsselt.

Die Sicherheit des beim Amazon Kindle eingesetzten DRM-Systems beruht also allein auf dem Hauptschlüssel der Geräte, der sich anfangs in der Geräteübersicht auf der Internetseite Amazons nachlesen ließ, später aber aus dieser Übersicht entfernt wurde.

Verlauf

Am 12. Dezember 2007 veröffentlichte der bei der auf Reverse-Engineering spezialisierten Firma Hex-Rays S.A. arbeitende Entwickler Igor Skochinsky ein Programm, das die Gerätenummer eines Kindle ausliest und daraus den für die Entschlüsselung einer DRM geschützten Mobipocket-Datei nötigen PID erzeugt [42]. Er gab an, das Programm geschrieben zu haben, um damit den PID von Mobipocket-Büchern ändern und diese damit auf dem Kindle lesen zu können.

Anfang 2008 veröffentlichte dann eine Person mit dem Pseudonym „The Dark Reverser“ die Software MobiDeDrm, die es ermöglicht, DRM geschützte Mobipocket-Dateien zu entschlüsseln [37]. Am 17. Dezember 2009 veröffentlichte eine Person mit dem Pseudonym „i♡cabbages“ die Software unswindle, die eine grafische Oberfläche bietet [23].

Am 29. Dezember 2009 veröffentlichte eine Person mit dem Pseudonym „Comprehensive Mazama Book DRM with Topaz Cryptography“ (CMBTC) eine gleichnamige Software, die Topaz-Dateien entschlüsseln kann [7]. Daraufhin äußerte sich Joshua Shagam, ein Software-Entwickler, der bis 2007 bei Amazon an dem Topaz-Format gearbeitet hatte, am 7. Januar 2010 zu dieser Software [40]. Shagam gab an, die Verschlüsselung des Topaz-Formats wurde unter großem Zeitdruck und der Vorgabe entwickelt, die Verschlüsselung dürfe keine externen Bibliotheken benutzen und ihr Speicherbedarf und Rechenaufwand müsse möglichst gering sein. Er sei eigentlich davon ausgegangen, dass die Verschlüsselung schon nach Wochen oder Monaten geknackt würde und es habe ihn verwundert, dass es zwei Jahre gedauert hätte und Amazon innerhalb dieses Zeitraums keine Anstrengungen unternommen habe, die Verschlüsselung zu verbessern. Trotzdem die Verschlüsselung sehr schwach sei und eine große kryptografische Schwachstelle, die nicht nur ihm, sondern

[16] Die Chiffre wurde nie einem Peer-Review oder einer Kryptoanalyse unterzogen. Es ist also unsicher, ob die Chiffre überhaupt ernsthaft zur Verschlüsselung brauchbar ist. Gemäß einer Analyse von Andreas Muegge, genügt es auf Grund der Zusammensetzung der Gerätenummer 2^{36} Schlüssel durchzuprobieren [34], was die Angreifbarkeit der Chiffre an sich für den Kindle ohnehin irrelevant macht.

auch einem anderen anonymen Entwickler, der ihn kontaktiert und auch bereits einen Brute-Force-Decoder geschrieben hätte, bekannt sei, sei der Verschlüsselungsalgorithmus an sich nicht das Hauptproblem des Topaz-Formats. Vielmehr sei das Schlüsselverteilungsverfahren, welches dem Benutzer den Geräteschlüssel über Kindle4PC sehr einfach zugänglich mache, die wichtigste Schwachstelle. Ein sicheres Schlüsselverteilungsverfahren sei aber unmöglich.

Vertragsbedingungen

In den Vertragsbedingungen zum Kindle schließt Amazon jegliche Haftung aus und behält sich vor, ihre Dienste ohne Entschädigung des Nutzers einzustellen. Des Weiteren geschehen Vertragsänderungen automatisch nach Ermessen von Amazon zu jeder Zeit und werden im Kindle Store angekündigt, eine Weiternutzung bestätigt den Vertrag. Das Reverse-Enigneering und die Modifikation der Hardware stellen dabei einen Vertragsbruch dar.

Einige der Klauseln sind nach deutschem Recht sicherlich ungültig. Der Vertrag ist allerdings, auch wenn es eine deutsche Übersetzung gibt, auf die USA ausgelegt, da der Kindle in Deutschland nicht auf dem Markt ist und nur importiert werden kann.

Bewertung

Ähnlich wie bei CSS ist die Chiffre des MOBI- und Topaz-Formats zwar schwach, aber nicht entscheidend dafür, dass der Kopierschutz des Kindles umgangen wurde, denn dieser wurde durch Reverse-Engineering von Kindle4PC umgangen.

Anders als bei CSS hat Amazon die Verbreitung der Software, die das DRM-System des Kindles unwirksam macht, nicht zu verhindern versucht und so DRM Gegnern, wie der FSF, die den Kindle mehrfach kritisiert hat, nicht die Möglichkeit wie bei CSS gegeben, sich medienwirksam zu äußern. Jedoch finden sich auch so gut wie keine unrechtmäßigen Kopien von Büchern im AZW- oder Topaz-Format, was wahrscheinlich darauf zurückzuführen ist, dass PDF, DJVU und EPUB weit verbreitet sind und daher kein Interesse an Amazons proprietären Dateiformaten besteht.

Bemerkenswert und vielleicht zukunftsweisend sind die restriktiven Vertragsbedingungen. Vielleicht werden sich Kunden dann darauf einstellen müssen, dass digitale Inhalte in gewisser Weise nur noch ausgeliehen werden können und kein wirklicher Anspruch mehr auf die Inhalte besteht.

4.4 Ubisofts Onlinekopierschutz

Ubisofts Onlinekopierschutz ist ein für Computerspiele der Firma Ubisoft Entertainment S.A. entwickeltes DRM-System, das unrechtmäßige Kopien von Computerspielen verhindern soll.

Funktionsweise

Der Onlinekopierschutz selbst ist ein Cloud-Computing-Dienst, bei dem der Kunde ein Computerspiel im Internet kauft, dies mit seinem Benutzerkonto assoziiert wird und er das Spiel schließlich auf beliebig vielen Computern installieren kann. Der bei Computerspielen sonst verbreitete Kauf eines Speichermediums und die Eingabe eines Lizenzschlüssels entfallen.

Um das Spiel zu spielen, muss der Benutzer eine Internetverbindung haben und dauerhaft mit dem Internet verbunden sein, denn bei Start des Spiels wird zunächst ein automatisches Update des Spiels durchgeführt. Während des Spiels besteht ständig eine Verbindung zu den Servern von Ubisoft, die sicherstellen soll, dass das Spiel nur einmal pro Benutzerkonto gleichzeitig gespielt wird. Wird die Internetverbindung länger als nur einige Sekunden unterbrochen, hält das Spiel an und weist den Benutzer darauf hin, die Internetverbindung wieder herzustellen. Kann dieser die Internetverbindung nicht in einem vorgebenen Zeitraum wiederherstellen, so wird das Spiel

beendet und der Benutzer innerhalb des Spiels an den letzten Checkpoint zurückversetzt, an dem er dann bei vorhandener Internetverbindung und Neustart des Spiel weiterspielen kann.

Verlauf

Der Onlinekopierschutz wurden bei den Spielen „Assasin's Creed 2“, das am 4. März 2010 erschienen ist[17], und „Silent Hunter 5“, das am 5. März erschienen ist, erstmals verwendet.

Bereits am 3. März solle der Kopierschutz beider Spiele geknackt worden sein [47] und es solle möglich gewesen sein, das Spiel auch ohne Internetverbindung spielen zu können. Am 7. März 2010 waren die Ubisoft Server, die zum Spielen der Spiele notwendig sind, nicht erreichbar [48]. Ubisoft gab daraufhin an, der Ausfall sei auf eine Distributed-Denial-of-Service-Attacke zurückzuführen [30]. Bei dem auch mit dem Onlinekopierschutz veröffentlichten Spiel „Die Siedler 7“ kam es einen Monat später zu ähnlichen Ausfällen [50].

Die hohe Ausfallrate und die allgemeine Funktionsweise des DRM-Systems führten bei vielen Spielern zu Frustration und Unbehagen. In vielen Foren, darunter auch Ubisofts eigenen Foren, wurde das Thema viel diskutiert und es häuften sich Beschwerden, die in Ubisofts Forum ignoriert wurden. Auch Redakteure von Computerspielezeitschriften und -blogs waren mit dem System sehr unzufrieden, so dass Ubisofts Onlinekopierschutz schnell ein sehr negatives Ansehen hatte. Beim Onlinehändler Amazon beispielsweise erreichte „Assasin's Creed 2“ über 120 Bewertungen mit einem von fünf Sternen, so viel wie kaum ein anderes Produkt, und wurde durch die Kunden mit negativen Schlagzeilen versehen. In einer Petition an Ubisoft [55] forderten über 14000 Personen schließlich die Einstellung des Kopierschutzes, da eine Internetverbindung für ein Einzelspielerspiel nicht notwendig sein dürfe, DRM in der Vergangenheit auch nicht zur Reduktion von unrechtmäßigen Kopien, sondern nur zur Frustration der Benutzer geführt hätte und der Online Kopierschutz nur zu einem Schaden für das Ansehen des Unternehmens führe. Auch ein Wiederverkauf wäre, da das Spiel an ein Benutzerkonto gebunden ist, sehr schwierig.

In den Nutzungsbedingungen der Spiele (EULA) [53], denen der Benutzer bei Installation zustimmen muss, erklärt dieser sich aber damit einverstanden, dass zum Spielen des Spiels eine Internetverbindung nötig ist und Ubisoft keine Haftung bei einem Serverausfall, Softwarefehlern oder dem Verlust von Benutzerdaten übernimmt. Ubisoft erklärt sich darin des Weiteren bereit, einen Patch zur Verfügung zu stellen, der das Spiel auch ohne Internetverbindung spielbar macht, sobald die Server abgeschaltet werden.

Für das am 7. September 2010 erschienene Spiel „R.U.S.E“, das anfangs auch mit dem Onlinekopierschutz erscheinen sollte, entschied sich Ubisoft jedoch für das vom Konkurrenten Valve entwickelte System Steam [49]. Im Gegensatz zu Ubisofts Lösung wird bei diesem nur eine Internetverbindung zum Herunterladen und zur Akitivierung eines Spiels benötigt, so dass der Spieler – anders als bei Ubisofts Onlinekopierschutz – nicht während des Spiels mit dem Internet verbunden sein muss [10]. Trotz der Bindung eines Spiels an ein Benutzerkonto und den damit genau wie beim Onlinekopierschutz verbundenen Schwierigkeiten beim Wiederverkauf ist Steam sehr verbreitet und wird mittlerweile von der Zielgruppe akzeptiert [11]. Sehr warscheinlich wird der durch den Onlinekopierschutz entstandene Ansehensschaden oder ein Umsatzrückgang Ubisoft zum Wechsel des DRM-Systems bewegt haben, so dass die Kritik der Kunden nicht zur Einstellung von DRM für Ubisoft Spiele, sondern zu einem Kompromiss geführt hat, der bei vielen Spielen anderer Hersteller bereits funktioniert [49].

Bewertung

Genau wie bei fast allen DRM-Systemen beruhte die Sicherheit von Ubisofts Onlinekopierschutz auf der Annahme, dass, da der Quelltext der Spiele nicht vorläge, eine Modifikation und damit Deaktivierung des DRMs nicht möglich sei. Diese Annahme hat sich aber seit Aufkommen erster

[17] Die Erscheinungsdaten unterscheiden sich zwischen Ländern und Kontinenten.

Kopierschutzmechanismen in der Computerspieleindustrie immer wieder als falsch herausgestellt, da Reverse-Engineering solcher Mechanismen für geübte Programmierer keine Herausforderung darstellt, was auch an Ubisofts Onlinekopierschutz, dessen Anforderung ständig mit dem Internet verbunden zu sein eine Neuerung war, aber trotz dieser größeren Herausforderung noch vor Erscheinen des Spiels geknackt wurde.

Wäre ein Computer mit Microsoft Windows ein von Stefik vorgeschlagenes ideales Trusted System, so hätte dies sicherlich ein Reverse-Engineering und eine Modifikation des Spiel verhindern können.

5 Schlussfolgerung und Ausblick

Nach mehr als einem Jahrzehnt Digital Rights Management hat sich die Vision Stefiks nicht wirklich bewahrheitet. Heutige DRM-Systeme sind technisch weit von Stefiks Konzept eines Trusted Systems entfernt, entziehen sich jeglicher sozialer Verantwortung und sind überhaupt nicht interoperabel und nachhaltig.

Das Konzept eines Trusted Systems ist grundlegend für jedes DRM-System, denn es macht, unter der Annahme, dass solche Systeme nicht manipulierbar sind, den Fehlgebrauch von Kryptografie, der DRM-Systemen zumeist zu Grunde liegt, möglich [44]. Der Fehlgebrauch besteht darin, dass DRM-Systeme Verschlüsselung, eine Technologie, die dafür gedacht ist, Nachrichten nur von Sender und Empfänger lesbar zu machen, benutzen, um digitale Inhalte für den Benutzer ohne das System nicht lesbar zu machen. Der Benutzer ist also in einer Situation Empfänger der Nachricht, der mittels des DRM-Systems im Besitz des Schlüssels ist, und in der anderen Situation Angreifer des Verschlüsselungssystems, dem es nicht möglich sein soll, die gespeicherte oder übermittelte Nachricht zu entschlüsseln, der aber gleichzeitig in Besitz des Geheimtextes (digitaler Inhalt des DRM-Systems) und des Schlüssels (wenn auch oft nicht unverschleiert) ist, die beide bestenfalls nur durch Hardware vor dem Auslesen geschützt werden. Diese Tatsache macht das kryptografische System unwirksam, da es nicht auf der Geheimhaltung und Nichtverfügbarkeit des Schlüssels, sondern nur auf dessen Verschleierung oder Speicherung in Hardware basiert [9, 16].

In der Realität wird es auch kein perfektes Trusted System geben und auch Hardware wird angreifbar sein, auch wenn die Angriffe schwieriger sein werden, da der Aufwand der Analyse deutlich höher ist und spezielleres Wissen und teureres Werkzeug erfordert. Im Bereich des Reverse-Engineering von integrierten Schaltungen wird es sicher in den nächsten Jahren im privaten und wissenschaftlichen Umfeld[18] noch bedeutenden Fortschritt und Automatisierung geben[19], so dass ein solches heute noch teures und aufwändiges Reverse-Engineering in Zukunft billiger und einfacher sein wird.

Des Weiteren ist oft noch nicht einmal ein Reverse-Engineering und eine Manipulation der Hardware notwendig, um ein Trusted System anzugreifen, denn kein komplexes System kann vollständig in Hardware implementiert werden, und es ist viel wahrscheinlicher, dass höhere und komplexere in Software implementierte Schichten fehlerhaft sind.

[18] Auch wenn keine offiziellen Angaben über kommerziell betriebenes Reverse-Engineering von integrierten Schaltungen existieren, gehe ich auf Grund der Vielzahl von Firmen, die auf Reverse-Engineering spezialisiert sind, davon aus, dass dies heute von allen größeren Herstellern betrieben wird.

[19] Zwar steigt die Strukturdichte bei Hochleistungsprozessoren und -speichern im oberen Marktsegment immer weiter, so dass immer teurere Maschinen zum Reverse-Engineering notwendig werden, jedoch ist eine solche Entwicklung bei einfacheren integrierten Schaltungen, wie beispielsweise in Smartcards, nicht so stark zu beobachten und die Strukturdichte solcher Schaltungen ist oft deutlich geringer, so dass ein Reverse-Engineering mit einfacheren Mitteln möglich ist.

Neben der Notwendigkeit, Aufnahmen einer integrierten Schaltung zu machen und Leitungen innerhalb dieser abzugreifen, ist auch die Software, die zur automatisierten oder teilautomatisierten Analyse von Schaltungen verwendet wird, bedeutend. Mit der Zeit wird diese auch immer besser, so dass gerade einfachere Schaltungen schneller verstanden werden können.

Der dritte wichtige Bestandteil einer solchen Entwicklung ist die Bildung einer Entwicklergemeinschaft, in der Wissen über das Thema durch informelle Bildung weitergegeben wird. In den letzten Jahren sind immer mehr Blogs und Vorträge im Netz zufinden, die nicht nur von Hobbyisten betrieben oder geschrieben wurden, sondern auch durch Mitarbeiter von Reverse-Engineering Firmen, welche die Geräte der Firmen in ihrer Freizeit nutzen.

Es ist also nur eine Frage der Zeit, bis Schlüssel eines DRM-Systems bekannt werden. Die Veröffentlichung solcher scheinbaren Geheimnisse erzeugt dabei auch eine manchmal große Aufmerksamkeit und Anerkennung. Mittlerweile hat sich das Knacken von DRM-Systemen schon zu einer Art Wettbewerb entwickelt, bei dem konkurriende Personen und Gruppen versuchen, als Erste das DRM-System zu knacken. Neue, immer schwieriger zu knackende Systeme, sind eine immer größere Herausforderung auch für Personen, die sich mit Kryptografie professionell beschäftigen (sowohl im geschäftlichen als auch wissenschaftlichen Umfeld) und keinerlei Interesse an den eigentlichen digitalen Inhalten besitzen, was letztendlich dazu führt, dass die DRM-Systeme noch schneller geknackt werden. Trusted Systems würden dabei eine noch viel größere Herausforderung darstellen, da diese noch viel schwieriger zu knacken sind.

Die Frage, *ob* ein DRM-System eine unrechtmäßige Nutzung von digitalen Inhalten verhindert, ist damit überflüssig und sollte vielmehr ersetzt werden durch die Frage, *wie lange* es eine solche Nutzung verhindert. Bei bisherigen DRM-Systeme dauerte letzteres fast immer nur Monate.

Gerade bei auf Hardware basierenden Systemen, wie sie zum Beispiel in der Filmbranche verwendet werden (DVD, HD DVD, Blu-ray Disc), setzt der Wechsel des DRM-Systems auch einen Wechsel der Hardware voraus[20], der vor allem für die Konsumenten, für die dies eine reine Ausgabe und keine Investition ist, sehr teuer wäre und selbst bei großem Druck durch die Produzenten, die nebenher zur Einführung einer neuen Technologie auch die alte Technologie, solange sie noch verbreitet ist, bedienen müssen, damit ihre Verkaufszahlen nicht einbrechen, mehrere Jahre dauern würde, also viel länger als das Knacken des Systems.

Zwar wird oft von Befürwortern von DRM behauptet, selbst wenn ein DRM-System geknackt worden sei, bedeute dies nicht, es sei unwirksam, denn der Durschnittsbenutzer, der den Großteil der Kunden ausmache, könne damit dann immer noch nicht umgehen. Diese Annahme ist falsch, wie sich gezeigt hat, denn sobald das DRM-System geknackt ist, genügt es, eine grafische Benutzeroberfläche zu programmieren, die lediglich das Auswählen von Dateien oder Medien erfordert, um das DRM-System auch von Durschnittsbenutzern umgehbar zu machen. Denn zur Benutzung solcher Software ist keinerlei Verständnis des DRM-Systems erforderlich [39].

In Zukunft wird es also, anders als von Stefik angestrebt, viele kurzlebige DRM-Systeme mit immer kürzer werdendem Lebenszyklus geben, denn das Knacken solcher Systeme wird immer einfacher und das folgende DRM-System muss eingeführt werden, sobald das vorherige DRM-System von einer bestimmten Anzahl von Benutzern umgangen werden kann. Zur Zeit könnte auch Trusted Computing diese Lebenszyklen nur herauszögern.

Es ist aber fraglich, ob Trusted Systems überhaupt eine Akzeptanz finden würden, denn der Versuch der Einführung von Trusted Computing hat großen Widerspruch gerade bei Bürgerrechtlern, Verbraucherschützern und Entwicklern und Nutzern Freier Software verursacht. Des Weiteren sind viele Menschen schon seit Einführung der Personal Computer daran gewöhnt, dass es sich dabei um Universalcomputer handelt, und es scheint nicht wahrscheinlich, dass eine solche große Änderung wie Trusted Systems kurz- bis mittelfristig Akzeptanz findet und sich nur durch Mangel an Alternativen durchsetzen könnte.

Eingebette Systeme werden hingegen als Geräte, die einmal gekauft spezifisch einen Zweck erfüllen, der sich ähnlich wie bei einfachen Gebrauchsgegenständen, nicht ändert, und sie werden deshalb vielmehr als einfacher Gebrauchsgegenstand wahrgenommen und vermarkt. Hier hat sich gezeigt, dass einige Benutzer sehr wohl bereit sind, die Einschränkungen von DRM zu akzeptieren, solange die Benutzung einfach gestaltet ist.

[20] Theoretisch könnte auch durch Update der Software auf den Geräten ein neues DRM-System bei gleichbleibender Speichertechnologie eingeführt werden, jedoch würde das bedeuten, dass sämtliche Hersteller an alle Kunden Softwareupdates verteilen müssten, was bis jetzt noch nie geklappt hat, wie beispielsweise an Microsoft Windows deutlich wird, an dem nur ein Hersteller beteiligt ist (von den Bedingungen, unter denen eingebette Systeme entwickelt werden und mit welchen Lebenzyklen in dieser Branche gerechnet wird, einmal ganz abgesehen). Des Weiteren könnten die verschiedenen DRM-Standards bei ein und demselben Medium – gerade bei technikuninteressierten Menschen – für Verwirrung, Unsicherheit und Frustration bei den Konsumenten führen. Zum anderen basiert das Geschäftsmodell der Hersteller solcher Geräte auch darauf, dass es immer wieder neue Technologien gibt, sobald der Markt gesättigt ist.

Freie Software, die unter der GPLv3 [21] veröffentlicht ist – das ist beispielsweise ein Großteil des GNU/Linux Betriebssystems, welches zur Zeit unabdingbar für das Funktionieren des Internets ist – wäre auch inkompatibel mit Trusted Computing, da die GPLv3 eine Klausel enthält, die dies verhindert. So würden DRM-Systeme lediglich auf proprietären Betriebssystemen laufen, aber gleichzeitig würde noch eine freie Alternative bestehen. Ein solches System könnte nie so umfassend sein wie von Stefik vorgesehen.

In Stefiks Vorstellung soll eine Interoperabilität und Lesbarkeit von Inhalten auch über Jahrzehnte gewährleistet sein. Bis jetzt wurde diese Interoperabilität jedoch nicht hergestellt. Bis auf größere Zusammenschlüsse, wie CSS, HD DVD oder Blu-ray Disc waren bisherige DRM-Systeme immer auf einzelne Firmen beschränkt, die ihre Technologien oft auch nicht weiter lizensierten. Denn durch die Bindung eines DRM-Systems an eine Firma ensteht ein Lock-in-Effekt, durch den die Kunden effektiv gebunden werden, da sonst ihre bisherigen digitalen Inhalten unlesbar würden. Des Weiteren bedeutet eine Lizensierung von DRM-Technologien auch immer eine Gefahr, denn für das Funktionieren der meisten DRM-Systeme ist Geheimhaltung, die durch Einbeziehen von immer mehr Personen immer schwieriger wird, notwendig[21].

Das Fehlen von Interoperabilität und langfristiger Lesbarkeit wird auch dazu führen, dass Inhalte, die ohne DRM oder Digitalisierung auch nach Jahrzehnten lesbar wären, nicht mehr lesbar sein werden und dass diese Inhalte mit der Zeit mehrfach gekauft werden müssen. Dies ist durchaus im Interesse der Inhalteanbieter, denn diese könnten durch den mehrfachen Verkauf größere Gewinne erzielen, sofern ihre Kunden diese Situation akzeptierten, und könnten aus diesem Grund auch Lebenszyklen von DRM-Systemen absichtlich verkürzen. Vermutlich wird die langfristige Lesbarkeit, wie sie bei Büchern besteht, für DRM-behaftete Inhalte nicht zu gewährleisten sein[22].

Blickwinkel bisheriger DRM-Systeme waren bis jetzt immer einzelne Firmen und Industriezusammenschlüsse, deren Ziel die Profitmaximierung durch Verhinderung jeglicher unrechtmäßiger Nutzung ist[23]. Diesem Gedanken liegt eine Vorstellung ihrer Kunden zu Grunde, in welcher der Kunde als potenzieller Krimineller[24] gesehen wird, bei dem jederzeit damit zu rechnen sei, dass er gegen die Nutzungsrechte verstoßen wolle. Diesem Misstrauen gegenüber dem Kunden soll der Kunde aber wiederum Vertrauen in proprietäre Technologien mit (bis heute) unbestimmter Zukunft und Lebensdauer entgegenbringen.

Durch das Misstrauensverhältnis zwischen Inhalteanbieter und Kunde kann DRM nicht als nachhaltiges Geschäftsmodell angesehen werden, denn es wird dadurch deutlich, dass die Kunden, sobald eine praktikable Alternative besteht, zu dieser wechseln würden.

Ein perfektes DRM-System ist also technisch und sozial unmöglich und DRM wird damit stets umgehbar sein. DRM ist des Weiteren weder ein langfristiges Geschäftsmodell noch werden digitale Inhalte und DRM-Systeme interoperabel oder langfristig lesbar sein. Es gibt auch keine Anzeichen, dass sich an dieser Situation etwas grundlegend ändern wird.

Auch wenn alternative Modelle wie Creative Commons oder Freie Software mit der Zeit immer

[21] Ausgenommen hiervon sind in gewisser Weise DRM-Systeme, die allen Teilnehmern eigene Schlüssel zuordnen, die selektiv widerrufen werden können. Ein Beispiel für ein solches System ist das AACS, das jedem Gerätehersteller eigene Schlüssel zuordnet, die durch Produktion neuer Medien widerrufen werden können. Allerdings zeigt das Bekanntwerden des High-bandwidth Digital Content Protection (HDCP) Hautpschlüssels, mit dem sich jeweils wieder Geräteschlüssel erzeugen lassen, dass auch solche Systeme den gleichen Problemen unterliegen, wenn auch auf anderer Ebene.

[22] Sowohl die Lagerung von Büchern als auch die langfristige Datenspeicherung setzen natürlich eine kontinuierliche Instandhaltung voraus.

[23] Mir sind keine neutralen Statistiken bekannt, die belegen, ob überhaupt ein finanzieller Schaden ensteht und wie hoch die Verluste durch unrechtmäßige Nutzung sind.

Die International Intellectual Property Alliance gibt die Verluste durch unrechtmäßige Downloads von Musikstücken in Deutschland mit 440 Mio. € an [1] und die Business Software Alliance gibt den Wert unrechtmäßig kopierter Software mit 1350 Mio. €, bei einer „Piraterierate“ von 28 % an [18]. Solche Zahlen aber scheinen voreingenommen und unrealistisch, wenn man bedenkt, dass die Werte nur Schätzungen und Hochrechnungen sind und die Herausgeber die Interessen der Firmen vertreten und durch solche Zahlen politische Veränderungen herbeiführen wollen.

[24] Diese Vorstellung kommt zum Beispiel in Begriffen wie „Softwarepiraterie“, „music piracy“ oder „Raubkopie“ und damit verbunden „Aufklärungsarbeit“ und Abmahnungen zum Ausdruck.

bedeutender werden, werden sie DRM und vor allem das damit verbundene Denkansatz nicht so schnell verdrängen können, denn die zukünftige Entwicklung von DRM ist eng mit dem viel weiter reichenden gesellschaftlichen Konflikt um Immaterialrechte verbunden, die im 21. Jahrhundert zusammen mit anderen Eigentumsrechten bestimmend sein werden.

Literatur

[1] International Intellectual Property Alliance, Hrsg. *2008 Special 301 Report. Special Mention Germany*. 2008. URL: http://www.iipa.com/rbc/2008/2008SPEC301GERMANY.pdf.

[2] Michael Arrington. *3 Million Amazon Kindles Sold, Apparently*. 29. Jan. 2010. URL: http://techcrunch.com/2010/01/29/3-million-amazon-kindles-sold-apparently/.

[3] Consumer Electronics Association, Hrsg. *Digital America – DVD*. 2005. URL: http://www.ce.org/Press/CEA_Pubs/929.asp.

[4] John Borland. *Program points way to iTunes DRM hack*. 24. Nov. 2003. URL: http://news.cnet.com/2100-1027_3-5111426.html.

[5] Steward Brand. „‚Keep Designing'". In: *Whole Earth Review* 46 (May 1985). *Tools and Ideas for the Computer Age*, S. 44–55. ISSN: 0759-5056. URL: http://www.wholeearth.com/issue/2046/.

[6] Bryan Chaffin. *iTMS DRM-Stripping Site (PlayFair) Pulled [Updated]*. The Mac Observer, Inc. 9. Apr. 2004. URL: http://www.macobserver.com/article/2004/04/09.12.shtml.

[7] CMBTC. *Kindle for PC DRM*. 29. Dez. 2009. URL: http://www.openrce.org/forums/posts/1199#3798.

[8] Peter Cohen. *PyMusique Author Hacks Apple's ITunes Fix. Once again, software will allow you to download songs without DRM*. MacCentral. 23. März 2005. URL: http://www.pcworld.com/article/120146/pymusique_author_hacks_apples_itunes_fix.html.

[9] Christian S. Collberg und Clark Thomborson. „Watermarking, tamper-proffing, and obfuscation: tools for software protection". In: *IEEE Trans. Softw. Eng.* 28.8 (2002), S. 735–746. ISSN: 0098-5589. DOI: http://dx.doi.org/10.1109/TSE.2002.1027797.

[10] Valve Corporation, Hrsg. *Offline Mode*. 2009. URL: https://support.steampowered.com/kb_article.php?ref=3160-AGCB-2555.

[11] Valve Corporation, Hrsg. *Steam & Game Stats*. 26. Sep. 2010. URL: http://store.steampowered.com/stats/.

[12] Ingemar Cox, Gwenaël Doërr und Teddy Furon. „Watermarking Is Not Cryptography". In: *Digital Watermarking*. Hrsg. von Yun Shi und Byeungwoo Jeon. Bd. 4283. Lecture Notes in Computer Science. 10.1007/11922841_1. Springer Berlin / Heidelberg, 2006, S. 1–15. URL: http://dx.doi.org/10.1007/11922841_1.

[13] Keith Dawson. *Apple Sends Cease-and-Desist To the Hymn Project*. Slashdot. 23. Feb. 2008. URL: http://news.slashdot.org/article.pl?sid=08/02/23/1915254.

[14] Whitfield Diffie und Susan Landau. *The Export of Cryptography in the 20th Century and 21st*. Sun Microsystems, Inc., 2005. URL: http://labs.oracle.com/people/slandau/export_control.pdf.

[15] Daniel Eran Dilger. *How FairPlay Works. Apple's iTunes DRM Dilemma*. 26. Feb. 2007. URL: http://www.roughlydrafted.com/RD/RDM.Tech.Q1.07/2A351C60-A4E5-4764-A083-FF8610E66A46.html.

[16] Cory Doctorow. *Microsoft Research DRM talk*. 17. Juni 2004. URL: http://craphound.com/msftdrm.txt.

[17] Eddie Edwards. *The Content Scrambling System (CSS). Disk Authentication*. URL: http://www.tinyted.net/eddie/css_auth.html.

[18] BSA Europe, Hrsg. *Softwarepiraterie in Deutschland klettert auf 28 %. Nur 19 Länder weltweit verzeichnen Anstieg der Piraterierate*. 11. Mai 2010. URL: http://portal.bsa.org/globalpiracy2009/pr/pr_germany.pdf.

[19] Derek Fawcus. *[Livid-dev] Time to help others?* 2. Okt. 1999. URL: http://web.archive.org/web/20000302153519/http://livid.on.openprojects.net/pipermail/livid-dev/1999-October/000397.html.

[20] Ken Fisher. *Apple's FairPlay DRM cracked*. Ars Technica. 5. Apr. 2004. URL: http://arstechnica.com/old/content/2004/04/3608.ars.

[21] Inc. Free Software Foundation, Hrsg. *GNU General Public License. Version 3*. 29. Juni 2007. URL: http://www.gnu.org/licenses/gpl.html.

[22] Bruce Houghton. *Amazon To Enter Crowded Download Market With MP3 Only Store*. 18. Dez. 2006. URL: http://hypebot.typepad.com/hypebot/2006/12/amazon_to_enter.html.

[23] i♡cabbages. *Circumventing Kindle For PC DRM (updated)*. 17. Dez. 2009. URL: http://i-u2665-cabbages.blogspot.com/2009/12/circumventing-kindle-for-pc-drm.html.

[24] Berkman Center for Internet & Society, Hrsg. *Informal DeCSS History Timeline*. URL: http://cyber.law.harvard.edu/openlaw/DVD/research/chronology.html.

[25] Steve Jobs. *Thoughts on Music*. Apple Inc. 6. Feb. 2007. URL: http://www.apple.com/hotnews/thoughtsonmusic/.

[26] Jon Lech Johansen. *[Livid-dev] confirmed: -CSS has been cracked-*. 23. Sep. 1999. URL: http://web.archive.org/web/20000823161342/http://livid.on.openprojects.net/pipermail/livid-dev/1999-September/000343.html.

[27] Auguste Kerckhoffs. „La cryptographie militaire“. In: *Journal des sciences militaires* (Jan. 1883), S. 5–38. URL: http://www.petitcolas.net/fabien/kerckhoffs/.

[28] *kindle development kit for active content*. Amazon.com, Inc. 2010. URL: http://www.amazon.com/kdk/.

[29] Butler W. Lampson. „A note on the confinement problem“. In: *Communications of the ACM* 16.10 (1973), S. 613–615. ISSN: 0001-0782. DOI: http://doi.acm.org/10.1145/362375.362389.

[30] John Leyden. *Ubisoft undone by anti-DRM DDoS storm. Protests over anti-piracy controls hobble games firm*. The Register. 8. März 2010. URL: http://www.theregister.co.uk/2010/03/08/ubisoft_anti_drm_hack_attack/.

[31] Derick Mains und Tom Neumayr, Hrsg. *Apple Unveils Higher Quality DRM-Free Music on the iTunes Store*. Apple Inc. 2. Apr. 2007. URL: http://www.apple.com/pr/library/2007/04/02itunes.html.

[32] Ka-Ping Yee Mark Miller und Jonathan S. Shapiro. *Capability Myths Demolished*. Techn. Ber. Johns Hopkins University, 2003. URL: http://srl.cs.jhu.edu/pubs/SRL2003-02.pdf.

[33] Eben Moglen. *The dotCommunist Manifesto*. Jan. 2003. URL: http://emoglen.law.columbia.edu/my_pubs/dcm.html.

[34] Andreas Muegge. *eBooks and DRM (2) – Secure Mobipocket Encryption*. 28. Juli 2008. URL: http://coderyder.wordpress.com/2008/07/28/ebooks-and-drm-2-secure-mobipocket-encryption/.

[35] Martin Persson und Alexander Nordfelth. *Cryptography and DRM*. Uppsala Universitet, 2008. URL: http://it.uu.se/edu/course/homepage/security/vt08/drm.pdf.

[36] Alex Pham. *Apple to wrap digital books in FairPlay copy protection [Clarified]*. 15. Feb. 2010. URL: http://latimesblogs.latimes.com/technology/2010/02/apple-ibooks-drm-fairplay.html.

[37] The Dark Reverser. *New Blog*. 13. Feb. 2008. URL: https://darkreverser.wordpress.com/2008/02/13/new-blog/.

[38] William Rosenblatt, Stephen Mooney und William Trippe. *Digital Rights Management: Business and Technology*. New York, NY, USA: John Wiley & Sons, Inc., 2001. ISBN: 0764548891.

[39] Bruce Schneier. *Crypto-Gram Newsletter. The Futility of Digital Copy Prevention*. 15. Mai 2001. URL: http://www.schneier.com/crypto-gram-0105.html#3.

[40] Joshua Shagam. *Interesting Topaz DRM development*. 7. Jan. 2010. URL: http://beesbuzz.biz/blog/e/2010/01/07-interesting_topaz_drm_development.php.

[41] Igor Skochinsky. *Hacking the Kindle part 3: root shell and runtime system*. 21. Dez. 2007. URL: http://igorsk.blogspot.com/2007/12/hacking-kindle-part-3-root-shell-and.html.

[42] Igor Skochinsky. *Mobipocket books on Kindle*. 12. Dez. 2007. URL: http://igorsk.blogspot.com/2007/12/mobipocket-books-on-kindle.html.

[43] *Source Code Notice*. Amazon.com, Inc. 2010. URL: https://www.amazon.com/gp/help/customer/display.html/ref=hp_left_cn?ie=UTF8\&nodeId=200203720.

[44] Paul Litva Spencer Cheng und Alec Main. *Trusting DRM Software*. Hrsg. von World Wide Web Consortium. Cloakware, Inc. Jan. 2001. URL: http://www.w3.org/2000/12/drm-ws/pp/cloakware.html.

[45] Mark Stefik. „Letting Loose the Light. Igniting Commerce in Electronic Publication“. In: *Internet Dreams. Archetypes, Myths, and Metaphors*. Hrsg. von Mark Stefik. MIT Press, 1996. ISBN: 0262193736. URL: http://www2.parc.com/istl/groups/uir/publications/items/UIR-1996-10-Stefik-InternetCommerce-IgnitingDreams.pdf.

[46] Mark Stefik. „Shifting the Possible: How Trusted Systems and Digital Property Rights Challenge Us to Rethink Digital Publishing“. In: *Berkeley Technology Law* 12.1 (1997). URL: http://www.law.berkeley.edu/journals/btlj/articles/vol12/Stefik/html/text.html.

[47] Peter Steinlechner. *Gerücht: Ubisofts Online-Kopierschutz geknackt. Erste Titel mit Zwangs-Online-DRM veröffentlicht*. Golem.de, Klaß & Ihlenfeld Verlag GmbH. 4. März 2010. URL: http://www.golem.de/1003/73593.html.

[48] Peter Steinlechner. *Offline: Ubisoft blamiert sich mit Online-Kopierschutz. DRM-Server waren stundenlang offline - Käufer verärgert*. Golem.de, Klaß & Ihlenfeld Verlag GmbH. 8. März 2010. URL: http://www.golem.de/1003/73682.html.

[49] Peter Steinlechner. *Ruse verwendet Steam statt Ubisoft-Launcher*. Golem.de, Klaß & Ihlenfeld Verlag GmbH. 12. Aug. 2010. URL: http://www.golem.de/1008/77156.html.

[50] Peter Steinlechner. *Ubisoft-Kopierschutz: Die Siedler 7 - offline über Ostern. Neue Gerüchte über angeblichen Hack des DRM-Systems von Ubisoft*. Golem.de, Klaß & Ihlenfeld Verlag GmbH. 8. März 2010. URL: http://www.golem.de/1003/73682.html.

[51] Frank A. Stevenson. *Cryptanalysis of Contents Scrambling System*. 11. Aug. 1999. URL: http://www.cs.cmu.edu/~dst/DeCSS/FrankStevenson/analysis.html.

[52] Jarrod Trevathan und Hossein Ghodosi. *Overview of Traitor Tracing Schemes*. 2003. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.5.5947.

[53] Igor Wallossek. *Installation: Mit der Lizenz für Spitzfindige*. Best of Media S.A. France. 16. März 2010. URL: http://www.tomshardware.de/Assassins-Creed-II,testberichte-240520-3.html.

[54] Wikipedia. *Amazon Kindle — Wikipedia, The Free Encyclopedia*. 2010. URL: http://en.wikipedia.org/w/index.php?title=Amazon_Kindle\&oldid=364546108#Original_Kindle.

[55] Ethan Woods. *No to Ubisoft DRM*. 2010. URL: http://www.petitiononline.com/ew15dl94/petition.html.

[56] Brecht Wyseur. „White-Box Cryptography“. Diss. 2009. URL: https://www.cosic.esat.kuleuven.be/publications/thesis-152.pdf.