Brian Winston

Caging the copycat. Wie neue Technologien eingeschränkt werden. Eine Fallstudie: Das Google Book Search Settlement

2010

https://doi.org/10.25969/mediarep/650

Veröffentlichungsversion / published version
Zeitschriftenartikel / journal article

Empfohlene Zitierung / Suggested Citation:
Winston, Brian: Caging the copycat. Wie neue Technologien eingeschränkt werden. Eine Fallstudie: Das Google Book Search Settlement. In: *Navigationen - Zeitschrift für Medien- und Kulturwissenschaften*, Jg. 10 (2010), Nr. 2, S. 85–94. DOI: https://doi.org/10.25969/mediarep/650.

Erstmalig hier erschienen / Initial publication here:
https://nbn-resolving.org/urn:nbn:de:hbz:467-5698

DIGITALE KODIERUNG UND REPRÄSENTATION

DVD, CSS, DeCSS

VON TILL A. HEILMANN

Die Logik von Digitalcomputern lässt sich anhand zweier, auf den ersten Blick gegensätzlicher, technischer Leistungen charakterisieren: der Wiederholbarkeit von Daten und ihrer Übersetzbarkeit. Beide ziehen für digitale Artefakte in Zweifel, was ›analogen‹ Werken gemeinhin als Wesensmerkmal zugesprochen wird: die Möglichkeit von Originalen bzw. ›echten‹ Kopien sowie die Identität eines Werkes mit sich selbst, insofern sie sich in dessen Erscheinung äußern kann. Die Folgen, welche digitale Kodierung für die Frage der Repräsentation im Computerzeitalter hat, will dieser Beitrag am Beispiel des digitalen Speichermediums DVD, des ›Kopierschutzes‹ CSS und dessen ›Hack‹ DeCSS illustrieren.

DIGITALCOMPUTER UND WIEDERHOLBARKEIT

Digitalcomputer sind Kopiergeräte.

Von der Informatik werden sie gemeinhin als ›symbolverarbeitende Maschinen‹ aufgefasst (Pflüger 2005: 70). Die zu verarbeitenden Symbole mögen z.B. Zahlen, Buchstaben oder Töne kodieren – je nachdem, ob ein Computer als Rechner, Schreibmaschine oder Musikbox verwendet wird. In jedem Fall schließt die Funktionsweise von Computern aber notwendig die Möglichkeit mit ein, die Symbole in identischer Form zu wiederholen, d.h. wieder und wieder zu speichern und zu übertragen. Das scheint bereits in dem theoretischen Modell auf, welches Alan Turing dem Computer noch vor dem Bau der ersten programmierbaren Rechenmaschinen gegeben hat. Turing (1936) beschreibt bekanntlich eine ›universelle Maschine‹, mit der sich sämtliche Zahlen berechnen lassen, die überhaupt berechnet werden können. Die hypothetische Maschine setzt Rechenvorgänge um, indem sie ein Papierband nach festgelegten Regeln schrittweise bewegt und solange Symbole davon liest und darauf schreibt, bis die zu berechnende Zahl endlich als Symbolkette auf dem Band geschrieben steht. Ein Rechenvorgang zerfällt so in eine Vielzahl kleinster Arbeitsschritte, die sich zu komplexeren und wiederkehrenden Abläufen ordnen, u.a. dem Löschen, Vergleichen und Vervielfältigen von Symbolketten: »These processes include copying down sequences of symbols, comparing sequences, erasing all symbols of a given form etc.« (Turing 1936: 235). Turings universelle Maschine ist daher mit verschiedenen Kopierroutinen ausgestattet, die unverzichtbare Bestandteile jeder Berechnung bilden.

Das Prinzip der Kopie als identischer Wiederholung findet sich nicht nur in der mathematisch-logischen Modellierung, sondern ist auch bestimmendes Merkmal der technisch-apparativen Implementierung von Digitalcomputern. Der amerikanische Literaturwissenschaftler Matthew Kirschenbaum (2008) hat kürz-

lich die Unterscheidung von forensischer und formaler Materialität zur Beschreibung digitaler Informationsverarbeitung vorgeschlagen. Kirschenbaum weist darauf hin, dass auf der physikalischen Ebene von Digitalcomputern jede Informationseinheit als einzigartige ›Einschreibung‹ in ein materielles Substrat existiert. So sind z.B. alle auf einer Festplatte aufgezeichneten, wenige Nanometer messenden magnetischen Flusswechsel, welche die Bits einer Datei speichern, aufgrund minimaler Verzerrungen ihrer Form unverwechselbare Spuren. Keine magnetische Markierung gleicht vollkommen einer anderen. Dasselbe gilt bei der Verarbeitung und Übertragung von Signalen für Spannungspegel in Schaltkreisen oder Kabeln. Forensisch gesehen ist jede materielle Einschreibung ein Unikat. Die Leistung von Digitalcomputern besteht darin, durch ein umfassendes technisches Regime aus Signalverstärkung und Fehlerkorrektur eine strikte Diskretisierung stabiler Zustände vorzunehmen, um auf der Grundlage ›verrauschter‹ und ›schmutziger‹ materieller Spuren ein scheinbar immaterielles Reich reiner Formen zu errichten, in dem die irreduzible Vielgestaltigkeit forensischer Spuren als Einförmigkeit rein formaler Identitäten und Differenzen behandelt wird. Ungeachtet seiner forensischen Einschreibung trägt jedes Bit den logischen Wert 0 oder 1 – niemals einen anderen, niemals etwas dazwischen. Alle gleichwertigen Bits tragen somit nicht nur den gleichen Wert; formal gesehen *sind* sie absolut gleich. Folglich ist jeder Akt der Speicherung, Übertragung und Verarbeitung digital kodierter Daten hinsichtlich deren formaler Materialität mit restlos identischem Ausgang wiederholbar.

> »Whereas forensic materiality rests upon the potential for individualization inherent in matter, a digital environment is an abstract projection supported and sustained by its capacity to propagate the illusion (or call it a working model) of *immaterial* behavior: identification without ambiguity, transmission without loss, repetition without originality.« (Kirschenbaum 2008: 11)

›Wiederholung ohne Eigenwilligkeit‹ ist das Kennzeichen digitaler Vervielfältigung.

Die Wiederholbarkeit, wie Walter Benjamin (1989) sie als technische Reproduzierbarkeit für die ›analogen‹ Medien Photographie und Kinofilm analysiert hat, erreicht mit Digitalcomputern also einen neuen techno-*logischen* Stand. Schwierig, wenn nicht gar unmöglich wird nicht nur der Begriff des Originals, sondern auch die Unterscheidung ›echter‹ Kopien (durch eine Zentralbank herausgegebene Banknoten etwa) von ›falschen‹. Weil Computer Daten symbolisch-diskret kodieren, statt sie mit kontinuierlichen Funktionen abzubilden, stellen alle ›Kopien‹ eines digitalen Artefaktes exakte Duplikate dar und sind, was ihren informationellen Gehalt anbelangt, prinzipiell nicht voneinander zu unterscheiden. Wiederholungen und – mehr noch – wiederholte Wiederholungen digitaler Daten, d.h. ›Kopien‹ von ›Kopien‹, kennen keine Qualitätseinbußen, wie sie für die Reproduktionen analoger Daten (überspielte VHS-Bänder oder photokopierte Texte beispielsweise) typisch sind. Digitale Information muss in ihrer Wiederholung nicht ›verrau-

schen‹. Daher sind Digitalcomputer nicht irgendwelche Kopiergeräte; sie sind gewissermaßen die *perfekten* Kopiergeräte. In Kombination mit der massenhaften Verfügbarkeit von Computertechnik, den stetig sinkenden Preisen für Speichermedien sowie den steigenden Übertragungskapazitäten elektronischer Datennetzwerke hat dieser technische Umstand für Computer als Medien weitreichende ökonomische Folgen – was vor allem Anbieter digitalisierter ›Medieninhalte‹ (Filme, Musik, Anwendungssoftware, Videospiele usw.) alarmiert, die gegen teures Geld verkaufen wollen, was potenzielle Käufer selbst so leicht und günstig vervielfältigen und verteilen können.

DAS CONTENT SCRAMBLE SYSTEM

Ein mustergültiger Fall für die Bestrebungen der Unterhaltungsindustrie, die unkontrollierte Verbreitung digitaler Daten technisch und rechtlich zu unterbinden, ist die DVD-Video, die Ende der 1990er Jahre die Nachfolge der VHS-Kassette antrat.[1] Die DVD-Spezifikation[2] beinhaltet verschiedene Verfahren zur Nutzungskontrolle von DVDs, darunter die *Regional Playback Control* (RPC) und das *Content Scramble System* (CSS). Aus dem Wissen darum geboren, dass digital kodierte Information prinzipiell verlustfrei kopierbar ist, stellen beide keine Kopierschutzmechanismen im herkömmlichen Sinne dar. Weder RPC noch CSS verhindern, dass der Inhalt einer DVD kopiert wird.[3] Beide Verfahren sind eher zum Komplex des Digital Rights Management (DRM), der sog. digitalen Rechteverwaltung, zu zählen, welche die kommerzielle Verwertbarkeit digitaler Daten durch strikte Beschränkung ihrer Nutzung garantieren soll (Grassmuck 2006). Während das RPC zur geographisch differenzierten Vermarktung von DVD-Titeln eine Einteilung der Welt in sechs große Regionen vornimmt und sich technisch gesehen einigermaßen

1 Die folgenden Ausführungen stützen sich wesentlich auf im Internet zugängliche Quellen, unter anderem auf die *DeCSS Central* (http://www.lemuria.org/DeCSS/, 27.02.2010), die *CSS Specifications Version 1.1* (http://cyber.law.harvard.edu/seminar/internet-client/readings/week2/02-08CSS.pdf, 27.02.2010); Gregory Kesdens *15-412 Operating Systems Lecture 33* (http://www-2.cs.cmu.edu/~dst/DeCSS/Kesden/, 27.02.2010), die FAQ-Liste des *Openlaw/DVD Forum* (http://cyber.law.harvard.edu/openlaw/DVD/dvd-discuss-faq.html, 27.02.2010) sowie die FAQ-Liste *DVD Demystified* (http://www.dvddemystified.com/dvdfaq.html, 27.02.2010).

2 Die *DVD Books*, welche die DVD-Spezifikation ausführen und von der DVD Format/Logo Licensing Corporation (DVD FLLC) herausgegeben werden, enthalten vertrauliche und rechtlich geschützte technische Informationen. Mit dem Kauf der Bücher wird daher ein Geheimhaltungsvertrag (*non-disclosure agreement*) abgeschlossen, der Käufern die Weitergabe der vertraulichen Informationen untersagt; siehe http://www.dvdfllc.co.jp/format/f_nosbsc.html, 27.02.2010.

3 Eigentliche Kopierschutzverfahren für DVDs sind etwa das *Analog Protection System* (APS) von Rovi – vormals Macrovision – gegen Videoband-Aufnahmen, das *Copy Generation Management System* (CGMS) sowie verschiedene ›Anti-Ripping‹-Mechanismen wie *ARccOS* von Sony, *RipGuard* von Rovi oder *ProtectDISC* von Protect Software. Nur APS und CGMS sind Teil der DVD-Spezifikation; alle anderen Techniken wurden unabhängig von dieser erst später entwickelt und umgesetzt.

simpel gestaltet, ist das CSS ein aufwändiges kryptographisches System, dessen Funktionsweise nun in seinen Grundzügen vorgestellt werden soll.

CSS ist eine symmetrische Verschlüsselungstechnik, die von Matsushita und Toshiba speziell für die DVD-Video entwickelt wurde (Marks/Turnbull 1999: 13). Obwohl von deren Spezifikation nicht zwingend vorgeschrieben, kommt sie bei so gut wie allen kommerziellen DVDs zur Anwendung. CSS erlaubt es, die mit Kompressionsalgorithmen digital kodierten Bild- und Toninhalte zusätzlich kryptographisch zu kodieren, um deren Wiedergabe durch unautorisierte Geräte zu verhindern. Nach den Plänen der von den großen Filmstudios eingerichteten DVD Copy Control Association (DVD CCA) darf allein Hard- und Software, die von ihr für eine »Verwaltungsgebühr« von jährlich 15.500 US-Dollars lizenziert wurde, den Datenstrom einer CSS-kodierten DVD dekodieren und wiedergeben können. Zu diesem Zweck installiert CSS ein in seiner Verschachtelung paranoisch anmutendes Verschlüsselungssystem mit ›Beglaubigungsschlüsseln‹ (*authentication keys*), ›Sitzungsschlüsseln‹ (*session keys*), ›Spielerschlüsseln‹ (*player keys*), ›Plattenschlüsseln‹ (*disk keys*), ›Titelschlüsseln‹ (*title keys*) und ›Bereichsschlüsseln‹ (*sector keys*). Entsprechend umständlich oder gar verwirrend mag die folgende Darstellung erscheinen.

Vereinfacht gesagt funktioniert die Wiedergabe einer CSS-kodierten DVD so, dass das Abspielgerät mit seinem Schlüssel die Schlüssel der DVD entschlüsselt, um mit diesen Schlüsseln wiederum die eigentlichen Bild- und Toninhalte zu entschlüsseln. Jedes CSS-lizenzierte Abspielgerät aus Hardware oder Software, d.h. jeder DVD-Player, aber auch jedes kommerzielle PC-Programm wie z.B. CyberLink PowerDVD, besitzt einen kleinen Satz an Spielerschlüsseln, welche dem Hersteller von der DVD CCA unter strengen Auflagen zugeteilt werden. Insgesamt gibt es 409 solcher Schlüssel, die selbstredend der Geheimhaltung unterliegen. Auf der anderen Seite besitzt jede CSS-kodierte DVD einen Plattenschlüssel, der auf ihr genau 410 Mal gespeichert ist: nämlich je einmal mit jedem der 409 möglichen Spielerschlüssel sowie ein weiteres Mal mit sich selbst verschlüsselt. Zusätzlich enthält die DVD einen sog. Hashwert des unverschlüsselten Plattenschlüssels, eine Art digitale ›Erkennungsmarke‹. Alle diese Informationen sind in einem geschützten Bereich der DVD abgelegt, auf den nur autorisierte Geräte zugreifen dürfen. Ist das abspielende Gerät keine Hardware mit speziellen Schaltkreisen zur CSS-Entschlüsselung (also kein ›regulärer‹ DVD-Player, der an einen Fernseher angeschlossen wird), sondern als Software implementiert, muss es sich daher zunächst gegenüber dem auslesenden Computerlaufwerk authentifizieren, was mithilfe des Beglaubigungsschlüssels geschieht. Im Rahmen dieses Vorgangs wird zudem ein temporärer Sitzungsschlüssel ausgehandelt, der den folgenden Datenaustausch zwischen Laufwerk und Abspielgerät verschlüsselt, damit die verschiedenen Schlüssel bei ihrer Übertragung nicht in unverschlüsselter Form abgefangen werden können. Soll die DVD wiedergegeben werden, muss das Abspielgerät deren Plattenschlüssel in Erfahrung bringen. Dazu liest es nach erfolgter Authentifizierung dessen 410 unterschiedlich verschlüsselte Instanzen aus und pro-

biert sie solange mit seinen Spielerschlüsseln durch, bis die Entschlüsselung gelingt. Dass der Plattenschlüssel korrekt entschlüsselt wurde, kann anhand seiner mit sich selbst verschlüsselten Instanz und seinem Hashwert verifiziert werden. Die Ton- und Bildinhalte wiederum sind mit einzelnen Titelschlüsseln verschlüsselt, welche ihrerseits durch den Plattenschlüssel verschlüsselt auf der DVD gespeichert sind. Als nächstes werden daher der entschlüsselte Plattenschlüssel und die noch unverschlüsselten Titelschlüssel an das Abspielgerät übermittelt (was wohlgemerkt alles durch den Sitzungsschlüssel *ver*schlüsselt geschieht). Daraufhin liest das Laufwerk nacheinander die in Bereiche unterteilten Daten des wiederzugebenden Titels aus und schickt sie an das Abspielgerät. Dieses entschlüsselt nun mit dem Plattenschlüssel den Titelschlüssel, damit den aktuellen Bereichsschlüssel und mit diesem schließlich die Bild- und Tondaten des jeweiligen Bereichs. Die eigentliche Ver- bzw. Entschlüsselung erfolgt, indem die zu en- bzw. dekodierenden Daten (also die verschiedenen Schlüssel wie auch die Bereichsdaten für Bild und Ton) Stück für Stück mit einer pseudo-zufällig erzeugten Bitfolge XOR-verknüpft werden.[4] Die logische XOR-Operation ist selbstinvers, d.h. ihre zweimalige Anwendung führt wieder zum Ursprungswert zurück, weshalb dieselbe Funktion für Ver- wie Entschlüsselung verwendet werden kann. Die Bitfolge, die dafür als Chiffrierstrom mit den Daten verknüpft wird, kommt durch Verschaltung eines Schlüsselpaars (z.B. des Titel- und des Bereichsschlüssel für die Kodierung der Bild- und Tondaten) mit zwei linear rückgekoppelten Schieberegistern (LFSR) zustande. Das Schlüsselpaar bildet die Anfangswerte der LFSRs, deren schrittweise Ausgabewerte miteinander verrechnet die pseudo-zufällige Bitfolge ergeben.

Ziel dieses mehrstufigen kryptographischen Verfahrens ist es offenkundig nicht, das Kopieren auf DVD gespeicherter Daten zu verunmöglichen. Vielmehr macht CSS den ›Besitz‹ kopierter Daten für all jene nutzlos, die nicht über ein autorisiertes Abspielgerät und den zu den Daten gehörigen Schlüssel verfügen. Einer privat angefertigten Kopie einer CSS-kodierten DVD aber fehlt eben dieser Plattenschlüssel, der im geschützten Lead in-Bereich der Scheibe gespeichert ist. Erschwerend kommt hinzu, dass brennbare Rohlinge – wenigstens des Typs DVD-R(W) – in diesem Bereich nicht beschrieben werden können. Auch wenn der Plattenschlüssel also bekannt wäre, könnte man ihn nicht mit auf die kopierte DVD geben. Kurz: Ohne den Spielerschlüssel eines von der DVD CCA lizenzierten Geräts und ohne den Plattenschlüssel einer industriell gepressten DVD sollten deren Bild und Ton nicht in unverschlüsselter Form zu haben sein.

DER ›HACK‹: DeCSS

Im Spätherbst 1999, gut zwei Jahre nach der Markteinführung der DVD-Video, waren die Sicherheitsmechanismen von CSS ausgehebelt. Jeder Computerbenut-

4 Somit ergeben eine 0 (des Datenstroms) und eine 0 (der pseudo-zufälligen Bitfolge) eine 0, 0 und 1 eine 1, 1 und 0 eine 1, 1 und 1 eine 0.

zer konnte nun mit geringem technischem Aufwand aber ohne autorisiertes Gerät und Plattenschlüssel alle CSS-kodierten Scheiben auf seinem Heimrechner abspielen und auch unverschlüsselt abspeichern. Möglich machten das verschiedene Computerprogramme, die aus Kreisen der Hacker-Subkultur und der Bewegung für freie und Open-Source-Software stammten.

In der journalistischen Berichterstattung wird das Knacken von CSS meist mit dem Norweger Jon Johansen in Verbindung gebracht. Johansen, zum Zeitpunkt des Geschehens knapp 16 Jahre alt, gehörte zu einer Gruppe von Hackern, die sich *Masters of Reverse Engineering* (MoRE) nannten. Im Oktober 1999 kündigte er auf der LiViD-Mailingliste zur Entwicklung eines freien Linux-Mediaplayers ein Windows-Programm namens DeCSS an, das alle CSS-kodierten DVDs würde entschlüsseln können. Kurz danach wurde dieses Programm von unbekannter Person im Quelltext (sozusagen der von jedermann einsehbaren und lesbaren ›Bauanleitung‹) wirklich auf der Mailingliste veröffentlicht und in Windeseile über das Internet in alle Welt getragen. Die Abspielsperre für unautorisierte Geräte war damit faktisch aufgehoben. Der genaue Ablauf der Ereignisse, die zu DeCSS und seiner weltweiten Verbreitung führten, bleibt bis heute unklar.[5] Als einigermaßen gesichert gelten darf, dass es MoRE sowie einem weiteren Hackerkollektiv mit Namen *Drink or Die* (DoD) im September 1999 gelang, die Funktionsweise von CSS aufzudecken und zu umgehen. Die entscheidende Information dazu soll angeblich ein unbekannter Hacker geliefert haben, der einen Spielerschlüssel aus der kommerziellen DVD-Software der Firma Xing extrahiert hatte. Noch vor dem Erscheinen von DeCSS brachte die Gruppe DoD das Programm DVD Speed Ripper heraus, das zunächst jedoch nicht alle geschützten Scheiben entschlüsseln konnte und auch nicht im Quelltext, sondern nur als ausführbare Datei verteilt wurde. Eine verbesserte Dekodierroutine, die mit sämtlichen DVDs funktionierte, floss wenig später von DoD über einen anonym bleibenden deutschen Hacker in die Entwicklung von DeCSS ein. Außerdem gelangten die Informationen zum Entschlüsseln CSS-kodierter Scheiben an die Gemeinde der Linux-Entwickler. Von diesen hatte der Engländer Derek Fawcus bereits einige Zeit zuvor den Authentifizierungsvorgang, der Zugriff auf den geschützten Bereich einer DVD gab, als Programm implementiert und im Quelltext veröffentlicht. All diese Anstrengungen kulminierten schließlich in DeCSS, das erstmals die bequeme Entschlüsselung einer DVD per Mausklick gestattete. Tatsächlich soll Johansens eigener Beitrag zum Windows-Programm, das ihm bald großen juristischen Ärger einbringen würde, lediglich in der Zusammenführung von Fawcus' Authentifizierungspro-

5 Siehe dazu u.a. die Stellungnahme von DoD und MoRE (http://www.lemuria.org/DeCSS/dvdtruth.txt, 27.02.2010), Frank A. Stevensons Erklärung vor Gericht (http://w2.eff.org/IP/Video/DVDCCA_case/20000107-pi-motion-stevensondec.html, 27.02.2010), das erstinstanzliche Osloer Gerichtsurteil im Fall Johansen (http://w2.eff.org/IP/Video/Johansen_DeCSS_case/20030109_johansen_decision.html, 27.02.2010) und die Chronologie der Ereignisse – mit inzwischen nicht mehr funktionierenden Links auf die entsprechenden Einträge der LiViD-Mailingliste – des *Openlaw/DVD Forum* (http://cyber.law.harvard.edu/DVD/research/chronology.html, 27.02.2010).

ramm und DoDs verbesserter Dekodierroutine unter einer graphischen Benutzeroberfläche bestanden haben.

Kaum war DeCSS im Quelltext bekannt und das Funktionieren von CSS damit offengelegt, meldete sich der Spielentwickler Frank A. Stevenson im Internet mit einer Kryptoanalyse des Verschlüsselungsverfahrens zu Wort.[6] Stevenson entdeckte fundamentale Designfehler, die CSS prinzipiell unsicher machten. Eine der entscheidenden Schwächen bestand in der geringen Länge der Schlüssel, die mit 40 Bit deutlich zu kurz waren.[7] Daher hielt CSS schon Ende der 1990er Jahre einer *brute force attack*, einem Angriff mit der geballten Rechenkraft eines zu jener Zeit handelsüblichen PCs nicht stand, welcher in weniger als einem Tag einfach alle 2^{40} möglichen Schlüssel durchrechnen konnte. Stevenson fand darüber hinaus schwerwiegende Mängel in der Erzeugung der pseudo-zufälligen Bitfolge, die als Chiffrierstrom dient. Die von den beiden LFSRs generierte Bitfolge war kryptographisch so schwach (so wenig ›zufällig‹ also), dass man von den Ausgabewerten auf die Anfangswerte der LFSRs und damit die zur Kodierung verwendeten Schlüssel schließen konnte. Auch der auf einer DVD gespeicherte Hashwert des Plattenschlüssels (dessen digitale ›Erkennungsmarke‹) erwies sich als anfällig für kryptoanalytische Angriffe, sodass aus ihm der Plattenschlüssel selbst rekonstruiert werden konnte, womit sich die Kenntnis eines funktionierenden Spielerschlüssels erübrigte. Im Oktober 1999 stellte Stevenson im Internet seine verschiedenen Methoden vor, anhand einer beliebigen CSS-kodierten DVD alle gültigen Spielerschlüssel – dieses von der DVD CCA so sorgsam gehütete Geheimnis – zu eruieren. Ein *reverse engineering* eines lizenzierten Schlüssels aus einer kommerziellen Software, wie es DeCSS erst möglich gemacht haben soll, war nicht mehr nötig. Hinfällig geworden war damit auch der Notfallplan der DVD CCA, kompromittierte Schlüssel ›zurückzurufen‹ und bei künftigen Geräten und Scheiben nicht mehr zu berücksichtigen.

Das Erscheinen von DeCSS und Stevensons daran anschließende kryptoanalytische Dokumentation lösten zugleich ein Problem, welches die Benutzer und Entwickler von Linux-Betriebssystemen lange umgetrieben hatte: Ende der 1990er Jahre gab es keine autorisierte DVD-Software für Linux und somit keine Möglichkeit, DVDs auf einem solchen System abzuspielen. Den etablierten Softwareproduzenten erschien der Markt für ein kommerzielles Programm zu klein und die für CSS jährlich anfallenden »Verwaltungsgebühren« waren restriktiv hoch. Dazu kam, dass die Linux-Entwicklergemeinde, die der Bewegung für freie und Open-Source-Software entstammte, nicht quelloffene Programme und Geheimhaltungsverträge, wie sie die Lizenzierung von CSS erforderte, grundsätzlich

6 Siehe Frank A. Stevenson: »Cryptanalysis of Contents Scrambling System«, 08.11.1999 (http://www.cs.cmu.edu/~dst/DeCSS/FrankStevenson/analysis.html, 27.02.2010).

7 Diese Wahl erfolgte u.a. mit Rücksicht auf die zu jener Zeit geltenden Exportbestimmungen der US-Regierung, die eine Ausfuhr von stärkeren Verschlüsselungsverfahren verboten oder zumindest erschwerten; siehe http://en.wikipedia.org/wiki/40-bit_encryption, 27.02.2010.

ablehnt. Mit dem neugewonnenen Wissen konnten die Entwickler nun aber Dekodierroutinen und Programme schreiben, die nicht auf ›gestohlene‹ Spielerschlüssel oder andere offensichtlich illegale Methoden zurückgreifen mussten und den Inhalt CSS-kodierter DVDs gleichwohl ohne lizenzierten Schlüssel wiedergeben konnten. Die bekannteste und heute am weitesten verbreitete Software dieser Art ist die Programmbibliothek libdvdcss, die wesentlich auf den Vorarbeiten von Fawcus und Stevenson aufbaut.[8] Um eine geschützte Scheibe abzuspielen, erzeugt libdvdcss zunächst selbständig eine Anzahl möglicher Spielerschlüssel. Sollten diese nicht funktionieren, startet es eine *brute force attack*, um den Platten- und zuletzt die Titelschlüssel zu brechen. Die Programmbibliothek ist weitgehend unabhängig von einzelnen Betriebssystemen und wird von vielen populären Open-Source-Programmen wie VLC Player, xine oder MPlayer zur Wiedergabe von DVDs verwendet.[9]

In seiner ursprünglichen Windows-Version ist das von der Industrie als ›Piraterie-Programm‹ verschrieene DeCSS wohl nie breitenwirksam zur praktischen Anwendung gekommen, schon gar nicht zum massenhaften ›Raubkopieren‹ von DVDs. Zwar konnte man den Inhalt einer DVD damit entschlüsseln, aber dieser Inhalt – mehrere Gigabyte große Dateien – ließ sich nicht leicht vervielfältigen und verbreiten. Geeignete Rohlinge hatten nur etwa die halbe Speicherkapazität einer kommerziellen Scheibe und waren zudem so teuer, dass sich der gesamte Aufwand für das Dekodieren, Komprimieren und Brennen einer Kopie im Vergleich zum Kauf eines Titels kaum lohnte. Festplattenspeicher war ebenfalls kostbar und ein damals handelsübliches Laufwerk mit zwei bis drei dekodierten Filmen bereits gefüllt. Und einer großflächigen Verbreitung über das Internet standen die gemessen an der Datenmenge eher geringen Übertragungsgeschwindigkeiten im Weg. Gewiss gab (und gibt) es ›Raubkopien‹. Diese wurden aber hauptsächlich von organisierten Gruppen aus dem südostasiatischen Raum gewerbsmäßig und mit professionellem Gerät hergestellt. Solche Kopien stellen vollständige 1:1-Duplikate kommerzieller DVDs dar, mitsamt der CSS-Verschlüsselung, Scheibenaufdruck und Verpackung. Auch war DeCSS keineswegs die erste technische Lösung, die Kontrollmechanismen von CSS zu umgehen und die Bild- und Tondaten in unverschlüsselter Form zu speichern. Zuvor schon hatte es eine erste Generation von ›Ripping‹-Programmen gegeben, die das Dekodieren einem autorisierten Gerät überließen und den von diesem entschlüsselten Datenstrom bei der Wiedergabe ›abfingen‹. Jenseits der Ermöglichung unverschlüsselter DVD-Kopien ist die Bedeutung von DeCSS vor allem darin zu sehen, dass sein Quelltext der Öffentlichkeit zum ersten Mal Einblick in die bis dahin weitgehend geheim gehaltene Funk-

8 Siehe die entsprechenden Hinweise im Quelltext der Datei css.c der Programmbibliothek, die von http://www.videolan.org/developers/libdvdcss.html (27.02.2010) heruntergeladen werden kann.

9 Die meisten großen Linux-Distributionen wie Debian, openSUSE oder Ubuntu werden aus rechtlichen Gründen ohne libdvdcss ausgeliefert, machen ihren Benutzern eine nachträgliche Installation der Programmbibliothek aber sehr leicht.

tionsweise von CSS bot und Entwicklern freier und Open-Source-Software erlaubte, Programme zur Wiedergabe geschützter Scheiben (seien diese ›legale‹ kommerzielle DVDs oder ›Raubkopien‹) für alternative Plattformen wie Linux-Betriebssysteme zu schreiben, ohne dabei die mit einer Lizenzierung durch die DVD CCA verbundenen finanziellen und rechtlichen Zugeständnisse machen zu müssen. Die historische Relevanz von DeCSS ist daher mehr theoretischer als praktischer Natur – mit sehr praktischen Auswirkungen allerdings.

Dass das Verschlüsselungsverfahren überhaupt gebrochen bzw. so problemlos umgangen werden konnte, dürfte drei Gründe haben: Erstens stellte bereits seine Umsetzung auch als Software ein beträchtliches Risiko dar. Software lässt sich ungleich einfacher analysieren als ein in die miniaturisierten Schaltkreise versiegelter Spezialchips gegossenes Hardware-Äquivalent. Sie kann mit Editoren eingesehen und – im Falle von DeCSS gar im ›menschenfreundlichen‹ Quelltext – ›gelesen‹, wenn nötig aus dem ausführbaren Objektcode in leichter verständliche Assemblerbefehle rückübersetzt und bei ihrem Vollzug im Arbeitsspeicher des ausführenden Computers ›beobachtet‹ werden. Dieser Umstand begünstigt ein *reverse engineering* der Technik, wie es mit der Player-Software von Xing geschehen und DeCSS in seiner ersten Version ermöglicht haben soll. Zweitens stellten sich entscheidende Teile von CSS (die verschiedenen Schlüssel, die Hashwerte der Plattenschlüssel und die den Chiffrierstrom erzeugenden LFSRs), nachdem sie in ihrer Software-Form analysiert worden waren, als mangelhaft implementiert heraus. Was zum dritten und wohl entscheidenden Punkt führt: CSS war als proprietäre Technik entworfen worden. Seine Mechanismen wurden von Matsushita und Toshiba geheim gehalten und nur Lizenznehmern unter Auflage höchster Vertraulichkeit bekannt gemacht. Dies verhinderte, dass es einer eingehenden Prüfung durch unabhängige Experten unterzogen werden konnte, wie es bei allen kryptographischen Systemen geschehen sollte. Der heute vielfach verwendete und als derzeit sicher eingestufte *Advanced Encryption Standard* (AES) beispielsweise durchlief einen mehrjährigen öffentlichen Begutachtungs- und Auswahlprozess. In einer scheinbar paradoxalen Verkehrung kann die Verlässlichkeit eines Verfahrens zum Schutz von Geheimnissen nach Kerckhoffs' Prinzip bzw. Shannons Maxime nur dann sichergestellt werden, wenn seine Funktionsweise vollständig offengelegt ist.

ELEKTRONISCHER ZIVILER UNGEHORSAM

Die Veröffentlichung von DeCSS im Internet zeigte kurz darauf die erwartbaren Folgen: Die DVD CCA und der Wirtschaftsverband der US-amerikanischen Filmstudios, die Motion Picture Association of America (MPAA), versuchten, die Bekanntmachung und Verbreitung des Programms rechtlich zu unterbinden. Die sich teilweise über Jahre hinziehenden juristischen Auseinandersetzungen zwischen den Interessensvertretern der Unterhaltungsindustrie und verschiedenen Privatpersonen sowie Fürsprechern der freien und Open-Source-Software können hier

nicht ausführlich dargestellt werden.[10] Es sollen nur einige Punkte aus dem rechtlichen und öffentlichen Geschehen rund um DeCSS herausgegriffen werden, die für Fragen der digitalen Kodierung und Repräsentation von Belang sind.

Nach dem ersten Erscheinen von DeCSS auf der LiViD-Mailingliste Ende Oktober 1999 wurde das Programm schnell ›weitergereicht‹ und auf Dutzenden von thematischen Webseiten angeboten. Der breiteren Öffentlichkeit blieb das nicht lange verborgen. Bereits am 1. November berichtete etwa Wired News unter dem Titel »DVD Piracy: It Can Be Done« über den DVD-›Hack‹.[11] Nun reagierte auch die MPAA und begann haufenweise Abmahnschreiben zu versenden, in denen die Betreiber entsprechender Webseiten aufgefordert wurden, DeCSS unverzüglich zu entfernen. Während einige der Forderung nachkamen, sahen sich zahlreiche andere durch das Vorgehen der MPAA in ihrer Überzeugung bestärkt, Information im allgemeinen und das Wissen um CSS bzw. DeCSS im Besonderen müsse frei verfügbar sein, und machten sich deshalb daran, das Programm auf möglichst viele Webseiten zu stellen und diese wiederum auf möglichst vielen anderen Seiten weltweit zu verlinken.

Der rapiden Verbreitung von DeCSS begegnete die DVD CCA, indem sie Ende Dezember vor dem Obergericht des Staates Kalifornien eine einstweilige Verfügung gegen fünfundzwanzig namentlich bekannte sowie fünfhundert weitere, ungenannt bleibende Webseitenbetreiber in den USA und anderen Ländern zu erwirken suchte.[12] Die Kläger behaupteten, das Programm verletze Betriebsgeheimnisse, da es durch *reverse engineering* der DVD-Software von Xing hergestellt worden sei (wofür sie allerdings keinerlei Beweise vorlegen konnten). Die Verteidigung – zwei von der Electronic Frontier Foundation (EFF) gestellte Anwälte – berief sich dagegen auf den 1. Zusatzartikel zur Verfassung der Vereinigten Staaten und das darin verbriefte Recht auf Meinungsfreiheit, welches die Veröffentlichung von DeCSS schütze. Nachdem das Gericht der einstweiligen Verfügung zwar nicht stattgegeben, in erster Instanz jedoch einen vorläufigen Rechtsschutz erlassen hatte, der die Veröffentlichung des Programms verbot, entschied das Appellationsgericht im November 2001 schließlich zugunsten der Angeklagten. Ausschlaggebend für den Entscheid war, dass das Gericht, gestützt auf einen Präzedenzfall, einen qualitativen Unterschied zwischen zwei verschiedenen Kodierungen von Computerprogrammen machte, dem Quelltext und dem Objektcode. Die Urteilsbegründung hält dazu fest:

10 Siehe dazu etwa das *Video and DVD Archive* der Electronic Frontier Foundation (http://w2.eff.org/IP/Video/DVDCCA_case/, 27.02.2010) oder das Archiv des *Openlaw/DVD Forum* (http://cyber.law.harvard.edu/openlaw/DVD/, 27.02.2010).

11 Siehe Andy Patrizio: »DVD-Piracy: It Can Be Done«, *Wired News*, 01.11.1999 (http://www.wired.com/science/discoveries/news/1999/11/32249, 27.02.2010).

12 DVD Copy Control Association, Inc. v. McLaughlin, No. CV 786804, 2000 WL 48512 (Cal. Super. Ct. Jan. 21, 2000); siehe http://cryptome.org/dvd-v-500.htm, 27.02.2010.

> »Like the CSS decryption software, DeCSS is a writing composed of computer source code which describes an alternative method of decrypting CSS-encrypted DVDs. Regardless of who authored the program, DeCSS is a written expression of the author's ideas and information about decryption of DVDs without CSS. If the source code were ›compiled‹ to create object code, we would agree that the resulting composition of zeroes and ones would not convey ideas. That the source code is capable of such compilation, however, does not destroy the expressive nature of the source code itself. Thus, we conclude that the trial court's preliminary injunction barring Bunner [Angeklagter, der Berufung eingelegt hatte; T.A.H.] from disclosing DeCSS can fairly be characterized as a prohibition of ›pure‹ speech.«[13]

Den Quelltext von DeCSS verstand das Appellationsgericht also als *Schriftstück*, das *Ausdruck* der *Ideen und Informationen* eines *Autors* sei – im Gegensatz zur ausführbaren Fassung des Programms (dem kompilierten Objektcode), welche lediglich eine Anordnung aus Nullen und Einsen darstelle und keine Ideen vermittle. Die *›reine‹ Rede* des DeCSS-Quelltexts hingegen, und damit auch dessen Wiedergabe auf der Webseite des Angeklagten Andrew Bunner, sei durch das Recht auf freie Meinungsäußerung geschützt.

Eine vergleichbares Verfahren der MPAA, das im Januar 2000 an einem New Yorker Bundesgericht gegen David Corley, den Betreiber der Hacker-Webseite 2600.com, wegen Veröffentlichung von DeCSS angestrengt wurde, ging weniger glücklich für den Angeklagten aus. Geklagt wurde hier nicht, wie in Kalifornien, wegen Verletzung von Betriebsgeheimnissen, sondern wegen Verstoßes gegen den *Digital Millennium Copyright Act* (DMCA). Mit der Beharrlichkeit und dem Einfallsreichtum der Hackergemeinde rechnend und wissend, dass Verfahren wie CSS überwunden werden können, hatte die Unterhaltungsindustrie in den 1990er Jahren darauf gedrängt, neue rechtliche Schranken zum Schutz ihrer wirtschaftlichen Interessen zu errichten (Marks/Turnbull 1999: 25). Bekanntester und folgenreichster Ausdruck dieser Bestrebungen ist eben der 1998 vom damaligen US-Präsidenten Clinton unterzeichnete DMCA, der vor allem den *Performances and Phonograms Treaty* (WPPT) der Weltorganisation für geistiges Eigentum (WIPO) von 1996 für das US-amerikanische Rechtssystem umsetzt.[14] Wichtige Teile beider Gesetze betreffen die Frage der sog. *anti-circumvention*: WPPT (Art. 18) und DMCA (Abs. 1201) legen fest, dass technische Maßnahmen zur Nutzungskontrolle rechtlich geschützter Werke, also etwa DRM-Verfahren wie CSS, nicht umgangen werden dürfen – wo technische Sperren vor technischen Angriffen versagen,

13 DVD Copy Control Association, Inc. v. Bunner, No. CV 786804 (Cal. App. Dep't Super. Ct. Nov. 1, 2001); siehe http://cryptome.org/dvd-v-bunner.htm (27.02.2010).

14 Siehe die Gesetzestexte in der US-Kongressbibliothek (http://thomas.loc.gov/cgi-bin/bdquery/z?d105:H.R.2281:, 27.02.2010) und bei der WIPO (http://www.wipo.int/treaties/en/ip/wppt/, 27.02.2010).

da sollen juristische Sperren helfen. Die Verteidigung berief sich, wie bereits im kalifornischen Gerichtsfall, auf die freie Meinungsäußerung, aber auch auf Ausnahmeklauseln im US-amerikanischen Urheberrechtsgesetz und im DMCA, die unter bestimmten Bedingungen eine nicht autorisierte Nutzung, ein *reverse engineering* und Kryptoanalysen geschützter Inhalte erlauben. Darüber hinaus zog die Verteidigung die Verfassungsmäßigkeit des DMCA überhaupt in Zweifel, weil sie durch diesen die vom 1. Zusatzartikel und dem Urheberrechtsgesetz garantierten Rechte verletzt sah. Trotz des Einsatzes und der Fürsprache zahlreicher angesehener Computerwissenschaftler wie Marvin Minsky, Ron Rivest und Harold Abelson gab das Gericht der MPAA Recht und untersagte Corley die Veröffentlichung von DeCSS sowie das Setzen direkter Weblinks darauf.

Aber nicht nur Personen, die DeCSS veröffentlichten und verteilten, hatten sich vor Gericht zu verantworten, sondern auch dessen Entwickler: Auf Betreiben der DVD CCA und der MPAA klagte die norwegische Staatsanwaltschaft im Jahr 2002 Jon Johansen, den einzig namentlich bekannten Schöpfer des Programms, nach dem norwegischen Strafgesetzbuch wegen ›unautorisierten Zugriffs‹ auf Computerdaten und -programme an. Weil gemäß norwegischem Recht jedoch bereits der Kauf einer kommerziellen DVD den Zugriff auf deren Daten autorisiert, das Anfertigen von Kopien für Privatzwecke nicht strafbar ist und Johansen selbst keine anderen illegalen Tätigkeiten nachgewiesen werden konnten, wurde er im Januar 2003 freigesprochen.[15]

Auf das juristische Vorgehen der Unterhaltungsindustrie antwortete die Hackergemeinde im Internet – unterstützt von Bürgerrechtsorganisationen wie der EFF – mit ›elektronischem zivilem Ungehorsam‹. Abgemahnte und verurteilte Webseitenbetreiber wie Corley entfernten DeCSS zwar aus ihren eigenen Angeboten; dafür verlinkten sie und hunderte weiterer Aktivisten systematisch andere Webseiten und solche außerhalb der USA, die das Programm immer noch bereithielten. Zudem ersannen sie Möglichkeiten, es in Formen zu kodieren und zu verbreiten, die ihrer Meinung nach von juristischen Verboten nicht berührt wurden. Der Informatiker David Touretzky, der beim Verfahren gegen Corley als Experte zugunsten des Angeklagten aussagte, richtete im Internet eine »Gallery of CSS Descramblers« ein, die verschiedene Varianten des DeCSS-Quelltexts vorstellte.[16] Mit den ›Ausstellungsstücken‹ der Galerie wollte Touretzky (2001) seine vor Gericht vertretene Position verdeutlichen, dass man keine klare Grenze zwischen Rede und Programm-Quelltexten ziehen könne und folglich auch diese durch den 1. Zusatzartikel geschützt seien. Nach der ersten Anhörung des Falles hatte das New Yorker Gericht – anders als das kalifornische – nämlich entschieden, Erläuterungen des Quelltexts von DeCSS seien durch das Recht auf freie

15 Siehe die engl. Übersetzung der Urteilsbegründung im *Video and DVD Archive* der Electronic Frontier Foundation (http://w2.eff.org/IP/Video/Johansen_DeCSS_case/20030109_johansen_decision.html, 27.02.2010).

16 Siehe David S. Touretzky: »Gallery of CSS Descramblers« (http://www.cs.cmu.edu/~dst/DeCSS/Gallery/, 27.02.2010).

Meinungsäußerung geschützt, nicht aber der Quelltext selbst, weil dieser in den ausführbaren Objektcode eines funktionsfähigen Programms kompiliert werden könne. Touretzkys Galerie enthält neben dem Quelltext des Programms in der Sprache C u.a. einen GIF-Screenshot davon, der als graphische Kodierung nicht kompilierbar ist, am Bildschirm aber alle im Quelltext enthaltene Information ›anzeigt‹. Eine andere Variante gibt den Quelltext in einem eigens für die Galerie erfundenen, formal rigorosen Dialekt von C wieder, für welchen jedoch kein Compiler existiert. Außerdem ›übersetzte‹ Touretzky die Funktionsweise von DeCSS bzw. dessen Quelltext Zeile für Zeile in schriftsprachliches Englisch. Zu guter Letzt präsentierte er den Besuchern seiner Webseite das Bild eines käuflich zu erwerbenden T-Shirts, auf welchem der Quelltext gedruckt stand. In der Folge steuerten Dritte weitere Varianten bei: DeCSS als Leseaufführung, als Strichcode, als Musicalnummer, als Haiku-Gedichte, als MIDI-Musikdatei, als Laufschrift-Animation in Star Wars-Manier, als ASCII-Art, als Beschreibung einer Hardware-Implementierung, als Yahoo-Grußkarte oder als in einer Bilddatei steganographisch verborgene Botschaft. Dazu kamen Umsetzungen in andere Programmiersprachen wie Scheme, Perl, Java, Javascript, PHP oder Pascal.

Die außergewöhnlichste Kodierung von DeCSS demonstrierte jedoch der Mathematiker Phil Carmody. Als Reaktion auf das Urteil des New Yorker Gerichts suchte er nach einer Darstellungsweise des Programms, die an sich, d.h. unabhängig von einer möglichen Verwendung zur Entschlüsselung CSS-kodierter DVDs, publikations- und archivierungswürdig war und deren Veröffentlichung daher nicht ohne weiteres verboten werden konnte. Ausgangspunkt von Carmodys Überlegungen war die Tatsache, dass sich ein Computerprogramm als Zahl begreifen lässt:[17] So kann man etwa die Bitfolge, welche die Reihe der Schriftzeichen kodiert, aus denen der DeCSS-Quelltext besteht, nicht nur als separate, 8-bittige ASCII-Kodenummern lesen, sondern ebenso gut als aufeinanderfolgende Stellen einer einzigen großen Zahl. Geeignete Kandidaten für archivierungs- und publikationswürdige Zahlen schienen Carmody Primzahlen zu sein, da Primalität ungeachtet etwaiger rechtlicher Bestimmungen eine grundlegende zahlentheoretische Eigenschaft mathematischer Objekte ist, und das Archiv, das er im Sinn hatte, waren die »Prime Pages«[18] des Mathematikprofessors Chris Caldwell, der im Internet verschiedene Listen von Primzahlen besonderer Art veröffentlicht. Carmodys Leistung bestand nun darin, eine Primzahl zu finden, die aufgrund ihrer Eigenschaften in eine von Caldwells Listen aufgenommen werden musste und dazu noch die Informationen des DeCSS-Quelltexts ›enthielt‹. Mit einigem mathematischem Geschick und der Rechenkraft seines PCs gelang es ihm, eine passende Primzahl mit 1905 Stellen auszumachen.[19] Weil diese Zahl erstens zum Zeitpunkt

17 Das gilt selbstredend nicht nur für Programme, sondern für alle digital kodierten Daten.

18 Siehe Chris Caldwell: »The Prime Pages. Prime number research, records, and resources« (http://primes.utm.edu/, 27.02.2010).

19 Für die mathematischen Einzelheiten siehe Phil Carmody: »The world's first illegal prime number?« (http://asdf.org/~fatphil/maths/illegal1.html, [19.03.2001], 27.02.2010) sowie

ihrer Entdeckung die zehntgrößte mithilfe des *elliptic curve*-Verfahrens aufgespürte Primzahl war, wurde sie in der entsprechenden Top 20-Liste von Caldwell publiziert. Zweitens kodierte sie auf ausgeklügelte Weise das DeCSS-Programm. Nimmt man die Binärdarstellung der Primzahl und entpackt sie mit dem weit verbreiteten Kompressionsprogramm gzip, ist das Resultat der Quelltext von DeCSS. Die von Carmody gefundene Zahl ist also selbst ein digitales ›Archiv‹ des für illegal befundenen Programms. Sie ist eine Primzahl *und* eine vollständige Darstellung des Entschlüsselungsverfahrens für CSS-geschützte DVDs. Einige Zeit später legte Carmody nach und präsentierte eine 1811-stellige Primzahl, deren Binärdarstellung zugleich der Objektcode eines CSS-Dekodierers für Linux-Betriebssysteme auf x86-Mikroprozessorarchitektur ist.[20] Diese Zahl muss nicht erst von einem Algorithmus wie gzip in eine andere überführt werden, sondern kann von entsprechenden Computern ›direkt‹ als Programm ausgeführt werden.

DIGITALCOMPUTER UND ÜBERSETZBARKEIT

Digitalcomputer sind Übersetzungsmaschinen.

Sie automatisieren En- und Dekodierprozesse von Symbolketten, indem Bitfolgen programmgesteuert zu neuen Bitfolgen ›umgeschrieben‹ werden. Beispiele für einen solchen weitgefassten Begriff von Kodierung als regelgeleiteter Übersetzung wurden verschiedene angeführt: die Übersetzung des Plattenschlüssels einer DVD in seinen Hashwert; der Titel- und Bereichsschlüssel in einen pseudozufälligen Chiffrierstrom; der CSS-verschlüsselten Bereichsdaten in unverschlüsselte Bild- und Tondaten; des ›menschenfreundlichen‹ Quelltexts von DeCSS in den ausführbaren Objektcode, aber auch eine Bilddatei, eine Musikdatei oder eine Primzahl; der Dezimaldarstellung einer Primzahl in deren Binärdarstellung; schließlich die Übersetzung der Binärdarstellung einer Archivdatei in den Quelltext eines Programms.

Die *Übersetzbarkeit* digital kodierter Information bildet gewissermaßen die Kehrseite ihrer Wiederholbarkeit. Die formale Materialität, wie Kirschenbaum sie durch Digitalcomputer verwirklicht sieht, bezeichnet nicht allein den Umstand absoluter Identität, d.h. vollkommener Einförmigkeit und eindeutiger Wertigkeit der informationstragenden Einheiten. Sie meint ebenso sehr (und noch mehr), dass diese Einheiten – die Bits – als typisierte Elemente mühelos und augenblicklich von einem Zustand in den anderen umgeschaltet werden können. Miniaturisierung, Menge und Geschwindigkeit der physikalischen Schaltungen erzeugen einen vermeintlich immateriellen Darstellungsraum, in welchem die von der Trägheit der Materie befreite Information beliebig formbar ist. Digitale Kodierung und Schalttechnik ermöglichen so ›endlose Permutationen‹ des Kodierten (Kirschen-

die entsprechende Seite von Caldwells »Prime Pages« (http://primes.utm.edu/glossary/xpage/Illegal.html, 27.02.2010).

20 Siehe Phil Carmody: »An Executable Prime Number?« (http://asdf.org/~fatphil/maths/illegal.html, [10.09.2001], 27.02.2010).

baum 2008: 145-146). Lautete die Botschaft des mechanischen Zeitalters und der technischen Reproduzierbarkeit ›Mehr vom selben‹, so ist das Versprechen der computerisierten Informationsgesellschaft und der digitalen Übersetzbarkeit ein alchemistisches ›Jedes Ding in jede Form‹.

An ihrer Übersetzbarkeit wird aber auch deutlich, wie problematisch die Frage danach ist, was digital kodierte Information ›eigentlich‹ repräsentiert. Bits *als* Bits, d.h. als Elemente formaler Materialität, gehören nicht der Ordnung des sinnlich Wahrnehmbaren an und müssen nach ihren übersetzenden Umschaltungen erst für menschliche Augen und Ohren (oder andere Sinne) transformiert werden – ein Vorgang, der seinerseits eine Übersetzung darstellt. Soll etwa der digital gespeicherte Quelltext eines Programms wie DeCSS für Menschen lesbar werden, müssen die Bitfolgen des Zeichensatzes, nach welchem die einzelnen Schriftzeichen kodiert sind, in die Bitfolgen der Glyphen einer Outline-Schrift wie TrueType übersetzt werden und diese wiederum in die Bitfolgen eines Bitmaps, welches dann in ein Leuchtmuster auf dem Bildschirm umgewandelt wird. Diese mehrstufigen Übersetzungen können sehr unterschiedlich ausfallen. Wer einmal ein aufwändig formatiertes Textdokument an einem Computer ›geöffnet‹ hat, auf welchem nicht alle zur gewünschten Darstellung notwendigen Schriften installiert sind, oder eine Webseite besucht, deren Zeichensatz vom Browser nicht ›richtig‹ erkannt wird, weiß, wie wandelbar die ›äußere Gestalt‹ digital kodierter Information ist.

Die Übersetzbarkeit von Computerdaten in verschiedene Repräsentationen betrifft jedoch nicht allein eher nebensächliche Aspekte wie die graphische Realisierung von Schriftzeichen. Sie stellt ganz grundsätzlich die ›Identität‹ digitaler Artefakte in Frage. Digital kodierte Information kann nie ›an sich‹, ›als solche‹ oder als ›sie selbst‹ erscheinen, sondern immer nur *in Übersetzung*. Dabei äußert sich nicht eine den Daten immanente ›Wahrheit‹. Die Übersetzung folgt einer Programmierung, deren potenziell unendliche Effekte durch Normen geregelt werden: Protokolle, Datenformate oder Zeichensätze wie HTTP, JPEG oder ASCII, und Anwendungssoftware wie Webbrowser, Bildbearbeitungsprogramme oder Texteditoren. Durch offene oder proprietäre Standards reglementiert (Galloway 2006) gelangen stets nur ausgewählte Übersetzungen von Computerdaten an die Benutzer. Vergleichbar der von Roland Barthes (1987: 13-14) beschriebenen Markierung einer Konnotation unter vielen als der scheinbar ›natürlichen‹ Denotation, wird üblicherweise eine Übersetzung digital kodierter Information als deren ›eigentliche‹ Repräsentation naturalisiert (Kirschenbaum 2008: 145-146). Die häufig getroffene Unterscheidung von ›kulturellen‹ Kodes wie denen der Poesie oder der Alltagssprachen und ›technischen‹ Kodes wie denen der elektronischen Datenverarbeitung, die u.a. in der juristischen Differenzierung zwischen der Rede des Autors im DeCSS-Quelltext und dessen verziffertem Objektcode ein Echo findet, ist also zumindest dann fragwürdig, wenn damit einem tendenziell endlosen Spiel von Bedeutungen auf der einen eine rigide Syntaktik ohne Freiheitsgrade auf der anderen Seite gegenübergestellt werden soll. Auch die Effekte ›techni-

scher‹ Kodierungen sind variabel und vom Kode selbst nicht völlig beherrschbar; auch die ver- und aufschiebende Bewegung der ›digitalen Differenz‹ (Tholen 1997) lässt sich nicht stillstellen.

Liegen das Vermögen und die Leistungsfähigkeit von Digitalcomputern in der unabschließbaren Übersetzbarkeit von Kodes begründet, dann ist die Tatsache, dass Computerdaten nach Belieben in neue Zustände umgeschaltet und d.h. eben verarbeitet werden können, wenigstens aus medientheoretischer Sicht nur deren uninteressantester Ausdruck. Das nichttriviale Moment der computerisierten Übersetzbarkeit von Daten besteht darin, dass digital kodierte Information eine irreduzible Vielzahl von Repräsentationen kennt, von denen keine als ihre ›richtige‹ oder ›authentische‹ identifiziert werden kann. Ein und dieselbe Webseite beispielsweise ist in sehr verschiedener Weise darstellbar: Man kann sie mit einem graphischen oder einem bloß textbasierten Browser öffnen, Schriftarten und -größen verändern, mit entsprechenden Erweiterungen zwischen mehreren Farbschemen auswählen, bestimmte Elemente (Werbebanner o.ä.) ausblenden oder sich den vom Browser verarbeiteten HTML-Quelltext anzeigen lassen. Alle daraus resultierenden Ansichten sind ›gültige‹ Darstellungen der Webseite.

Dasselbe gilt für einzelne Dateien, etwa eine JPEG-Datei: Bei der Arbeit am Computer erscheint sie dem Benutzer wahrscheinlich zuerst als Listeneintrag in einem Verzeichnisfenster mit Angabe ihres Namens, der Größe, des Typs und verschiedenen Zeitstempeln. Möglicherweise taucht sie auch als graphisches Icon oder als miniaturisierte Vorschau in einem Bildverwaltungsprogramm auf. Vielleicht ist der Benutzer nur an in die Datei eingebetteten Metadaten interessiert, welche die genauen Umstände der Bildaufnahme durch eine Digitalkamera beschreiben. Bildbearbeitungsprogramme können die Datei zudem als Histogramm darstellen, welches die Tonwertverteilung visualisiert. Selbstverständlich kann man in der Bilddarstellung Details vergrößern, so dass ein kleiner Ausschnitt den ganzen Monitor füllt. Und in besonderen Fällen können in der Datei steganographisch verborgene Informationen (z.B. der Quelltext von DeCSS) sichtbar gemacht werden. Für die Mehrzahl der Benutzer mag sich die Vollbilddarstellung einer JPEG-Datei wie deren ›eigentliche‹ Repräsentation ausnehmen; sie ist aber nur eine unter mehreren möglichen Übersetzungen.

Und noch denkbar einfach kodierte Daten, wie die in einer sog. reinen Textdatei gespeicherten, lassen sich nicht auf eine Darstellung reduzieren. Der ›Inhalt‹ auch einer solchen Datei kann in verschiedenen Schriftarten und -größen wiedergegeben werden, mit umgebrochenen oder trunkierten Zeilen, eingeblendeter Zeilennummerierung, Syntaxhervorhebung usw. Die Ausgabe ist jedoch keineswegs auf die graphischen Mittel des Alphabets beschränkt: Der Zeichensatz einer Textdatei, ASCII beispielsweise, beschreibt ja nicht die Gestalt von Schriftzeichen (die Glyphen), sondern deren Charaktere als Grapheme. Mit einem sog. Hexeditor kann man sich deshalb die numerischen Werte der Charaktere bzw. der einzelnen Bytes in hexadezimaler Notation anzeigen lassen. Noch einen Schritt weitergehend könnte man diese Bytes mit einem selbst geschriebenen kleinen Pro-

gramm oder Skript in die Binärdarstellung ihrer Bitwerte übersetzen, eine lange Folge von Nullen und Einsen. Nun mag man versucht sein, eben diese Zahlenwerte als ›richtige‹ Darstellung der Datei zu verstehen. Sind nicht die Zahlen die ›Essenz‹ der Datei, der Ursprung ihrer vielfältigen Übersetzungen? Aber schließlich stehen die Zahlen als Ziffern auf einem Bildschirm oder Blatt geschrieben. Auch sie sind das Produkt eines komplexen Übersetzungsvorganges, der im Lichtschein eines Monitors oder auf Papier fixiertem Farbstoff endet. Als formale Materialitäten existieren die Ziffern nicht. Im ›Innern‹ von Digitalcomputern, in Speicherchips und auf Festplatten, finden sich keine Ziffern, auch keine Zahlen, keine Nullen und Einsen. Was sich finden lässt (das notwendige mikroskopische Gerät vorausgesetzt), sind Spuren forensischer Materialität, die als formale Materialitäten erst programmgesteuert umgeschaltet und zuletzt in Repräsentationen übersetzt werden müssen, von welchen *eine* die auf einem Bildschirm oder Blatt sichtbare Folge der Ziffern 0 und 1 ist.

Die prekäre Identität digitaler Artefakte ist durchaus keine bloß akademische Angelegenheit. Als Amazon – um ein letztes Beispiel anzuführen – die zweite Version des *Kindle* auf den Markt brachte, konnte das Lesegerät digitale Bücher neu auch ›vorlesen‹, d.h. mittels einer synthetischen Stimme in gesprochene Rede übersetzen. In Umkehrung der phonozentrischen Tradition abendländischer Metaphysik behauptete die US-Autorenvereinigung, dies stelle eine unbefugte Tonaufführung der Bücher dar, ein laut Urheberrechtsgesetz vom ursprünglichen ›abgeleitetes Werk‹ (»derivative work«):[21] *Eine* Übersetzung (graphische Schriftzeichen auf dem E-Ink-Display) sollte die ›eigentliche‹, ›primäre‹ Repräsentation des Werkes sein, alle anderen dagegen (wie z.B. die stimmliche Sprachausgabe) davon ›abgeleitet‹ und ›sekundär‹. Technisch und juristisch kann die Übersetzbarkeit digital kodierter Information in bestimmte Darstellungen – im Falle der DVD durch CSS oder beim *Kindle* durch gezielte Deaktivierung der Sprachausgabefunktion – erschwert oder eingeschränkt werden. Die Frage aber, deren Problematik Carmodys Primzahlen-Bitfolgen-Dateiarchive-Computerprogramme deutlich zu machen suchen, bleibt in jedem Fall bestehen: Was ›bedeutet‹ digital kodierte Information? Was stellen digitale Artefakte dar? Die Antworten, welche die jeweiligen Repräsentationen darauf geben, sind nie nur Ergebnis ›neutraler‹ technischer Verfahren. Sie sind immer auch Resultate von Entscheidungen, die im weitesten Sinne computerpolitische genannt werden dürfen.

21 Siehe Michael Kwun: »Does the Authors Guild Want to Sue You for Reading Aloud to Your Kids?« (http://www.eff.org/deeplinks/2009/02/does-authors-guild-want-sue-you-reading-aloud-your, [11.02.2009], 27.02.2010).

LITERATURVERZEICHNIS

Barthes, Roland (1987): *S/Z,* Frankfurt a.M.: Suhrkamp.

Benjamin, Walter (1989): »Das Kunstwerk im Zeitalter seiner technischen Reproduzierbarkeit (Zweite Fassung)«, in: ders.: *Gesammelte Schriften*, Bd. VII, Frankfurt a.M.: Suhrkamp, S. 350-384.

Galloway, Alexander R. (2006): »Protocol vs. Institutionalization«, in: Chun, Wendy Hui Kyong/Keenan, Thomas (Hg.): *New Media, Old Media. A History and Theory Reader*, New York: Routledge, S. 187-198.

Grassmuck, Volker (2006): »Wissenskontrolle durch DRM: von Überfluss zu Mangel«, in: Hofmann, Jeanette (Hg.): *Wissen und Eigentum. Geschichte, Recht und Ökonomie stoffloser Güter*, Bonn: Bundeszentrale für politische Bildung, S. 164-186.

Kirschenbaum, Matthew G. (2008): *Mechanisms. New Media and the Forensic Imagination*, Cambridge, MA u.a.: MIT Press.

Marks, Dean S./Bruce H. Turnbull (1999): »Workshop on implementation issues of the WIPO Copyright Treaty (WCT) and the WIPO Performances and Phonograms Treaty (WPPT)«, http://www.lemuria.org/DeCSS/imp99_3.pdf, 27.02.2010.

Pflüger, Jörg (2005): »Wo die Quantität in Qualität umschlägt«, in: Warnke, Martin/Coy, Wolfgang/Tholen, Georg Christoph (Hg.): *HyperKult II. Zur Ortsbestimmung analoger und digitaler Medien*, Bielefeld: Transcript, S. 27-94.

Tholen, Georg Christoph (1997): »Digitale Differenz«, in: Warnke, Martin/Coy, Wolfgang/Tholen, Georg Christoph (Hg.): *HyperKult. Geschichte, Theorie und Kontext digitaler Medien*, Basel u.a.: Stroemfeld, S. 99-116.

Touretzky, David S. (2001): »Free Speech Rights for Programmers«, *Communications of the ACM*, Vol. 44, No. 8, S. 23-25.

Turing, Alan M. (1936): »On computable numbers«, *Proceedings of the London Mathematical Society*, Vol. 42, No. 2, S. 230-265.