Case No. 02A530

IN THE SUPREME COURT OF THE UNITED STATES OF AMERICA

DVD COPY CONTROL ASSOCIATION, INC., Applicant,

v.

MATTHEW PAVLOVICH, Respondent.

On Application for Stay of the Judgment of the Supreme Court of California

---

DECLARATION OF ALLONN E. LEVY IN OPPOSITION TO DVD CCA'S APPLICATION FOR STAY OF JUDGMENT OF THE SUPREME COURT OF CALIFORNIA

-- Jurisdiction Contested --

ARTHUR V. PLANK (Bar No. 072265)
ALLONN E. LEVY (Bar No. 187251)*
HOPKINS & CARLEY LLC
70 S. First Street
San Jose, CA 95113
Telephone: (408) 286-9800
Facsimile: (408) 998-4790
**Counsel of Record*

ORNAH LEVY (Bar No. 194683)
ATTORNEY AT LAW
5972 Spinnaker Bay Dr.
Long Beach, CA 90803
Telephone: (562) 961-6828
Facsimile: (562) 961-6838

Attorneys for Petitioner
Matthew Pavlovich

I, Allonn E. Levy, declare:

1. I am an attorney at law duly admitted to practice before all courts in California and am one of the attorneys representing Matthew Pavlovich, Defendant and Respondent.

2. Jurisdiction in this case is contested; as such this opposition and all supporting papers are not intended to constitute a general appearance.

3. The facts stated herein are true of my own knowledge and if called upon to do so, I could and would competently testify thereto. As to those matters stated on information and belief, I believe those matters to be true.

4. This declaration and supporting evidence is provided pursuant to United States Supreme Court rules 21 and 22. The evidence is provided to rebut the allegations made by DVD Copy Control Association Inc. (DVD CCA), Applicant herein.

5. I am informed and believe that at the time this action was commenced in December of 1999, Mathew Pavlovich was a student at the Purdue University in Indiana. Mr. Pavlovich is currently a resident of Texas and is employed in a small start-up company he founded. Pavlovich's current entrepreneurial efforts are unrelated to any aspect of this litigation.

6. While Matthew Pavlovich has never appeared in this action, on November 28, 2001, another defendant Andrew Bunner filed a motion for summary judgment in this action (DVD CCA v. McLaughlin, et al., Superior Court Case No. CV 786804). That motion has since been stayed by order of the trial court. The evidence filed with the trial court on behalf of Mr. Bunner is directly relevant to the issue before this Court in that it refutes the allegations of impending irreparable harm presented by DVD CCA.

7. Attached hereto as Exhibit A is a true and correct copy of the "Non-confidential Evidence In Support of Defendant Andrew Bunner's Motion For Summary Judgment" filed in

the trial court in this action (DVD CCA v. McLaughlin, et al., Superior Court Case No. CV 786804) on November 28, 2001.

8. The accompanying "Non-confidential Evidence In Support of Defendant Andrew Bunner's Motion For Summary Judgment" contains, as Exhibits 1-5, the declarations submitted to the trial court by Princeton Computer Science Professor Edward Felten (chief technical adviser to the U.S. Department of Justice in *United States v. Microsoft*), University of California-Berkeley Computer Science Professor David Wagner, Carnegie Mellon University Principal Computer Scientist Dr. David Touretzky, Carnegie Mellon University Computer Scientist Gregory Kesden, and Computer Scientist Roland Parviainen of Sweden's Luleå University of Technology. These declarations demonstrate that:

- DeCSS remains available, at the very least, at *hundreds* of locations on the Internet, in both source code and object code versions. Prof. Wagner Decl. ¶¶ 6-21; Dr. Touretzky Decl. ¶¶ 13-15, 18-23; Prof. Felten Decl. ¶¶ 14-15.
- Numerous additional programs performing the CSS descrambling function have been created in a variety of programming languages. Dr. Touretzky Decl. ¶¶ 14-15, 29; Prof. Wagner Decl. ¶¶ 22-25.
- In addition, DVD descrambling programs have been published in print by both MIT's journal *Technology Review* and *Wired Magazine*, and the *Wall Street Journal* published one of the CSS master keys. Dr. Touretzky Decl. ¶¶ 10, 29 & Exs. A, B, C.
- CSS and its algorithms and keys have been the subject of research, discussion, and teaching worldwide within the computer science community, both academic and non-academic. Prof. Felten Decl. ¶¶ 12-13, 16-21; Prof. Wagner Decl. ¶¶ 26, 28-33; Dr. Touretzky Decl. ¶¶ 14, 26-32; Kesden Decl. ¶¶ 1-8 & Ex. A; Parviainen Decl. ¶¶ 1-5.

- Other descriptions and representations of the CSS algorithms and keys have been created in a vast variety of formats. Cryptographer Frank Stevenson's technical paper describing the CSS algorithms and keys is widely known in cryptographic circles and is available on the Internet. Prof. Felten Decl. ¶¶ 17-20; Dr. Touretzky Decl. ¶¶ 11-12, 14-18, 28; Kesden Decl. ¶ 8 & Ex. B. Others have created narrative descriptions, mathematical descriptions, and graphical, animated, and musical renderings of the CSS algorithms and keys. Dr. Touretzky Decl. ¶¶ 14-18, 28.

9. Moreover, when requested in interrogatories to identify all sources currently disclosing the CSS algorithms or keys, DVD CCA frankly confessed it had given up any serious attempt to police its trade secrets. After identifying 72 of the "thousands of web sites and file transfer sites apparently claim[ing] to be posting materials containing Plaintiff's trade secrets," DVD CCA explained it was making no attempt to locate and identify, much less suppress, all of these thousands of sources publishing information about the CSS algorithms and keys, stating "Plaintiff cannot reasonably be expected to perform this process to verify the contents of thousands of web sites claiming to be posting Plaintiff's trade secrets." Plaintiff's Highly Confidential Supplemental Ans. & Objs. To Def. Andrew Bunner's First Set of Interrogs., at 3-5; Plaintiff's Highly Confidential Ans. & Objs. To Def. Andrew Bunner's First Set of Interrogs., at 3.[1]

10. Attached hereto as Exhibit B is a true and correct copy of the order of the California Supreme Court denying DVD CCA's Motion To Extend The Date Of Finality Of Decision, filed on December 24, 2002.

[1] This evidence was also submitted to the trial court on November 28, 2001 under separate cover. Should the

I declare under penalty of perjury under the laws of the State of California that the foregoing is true and correct.

DATED: 1/2, 2003

By:
Allonn E. Levy

court desire a copy of the filing, Respondent will provide same along with a motion to file said evidence under seal.

EXHIBIT A

RICHARD R. WIEBE (SBN 121156)
425 California Street, Suite 2025
San Francisco, CA 94104
Telephone: (415) 433-3200
Facsimile: (415) 433-6382

THOMAS E. MOORE III (SBN 115107)
TOMLINSON ZISKO MOROSOLI & MASER LLP
200 Page Mill Road, Second Floor
Palo Alto, CA 94306
Telephone: (650) 325-8666
Facsimile: (650) 324-1808

ALLONN E. LEVY (SBN 187251)
HS LAW GROUP
210 N. Fourth St. Suite 200
San Jose, CA 95112
Telephone: (408) 295-7034
Facsimile: (408) 295-5799

CINDY A. COHN (SBN 145997)
ROBIN D. GROSS (SBN 200701)
ELECTRONIC FRONTIER FOUNDATION
454 Shotwell Street
San Francisco CA 94110
Telephone: (415) 436-9333
Facsimile: (415) 436-9993

Attorneys for Defendant ANDREW BUNNER

SUPERIOR COURT OF THE STATE OF CALIFORNIA

COUNTY OF SANTA CLARA

DVD COPY CONTROL ASSOCIATION, INC.,
Plaintiff,
v.

ANDREW THOMAS MCLAUGHLIN; ANDREW BUNNER; et al.,
Defendants.

Case No. CV - 786804
DATE: January 29, 2002
TIME: 9:00 a.m.
DEPT.: 2
HONORABLE WILLIAM J. ELFVING

NON-CONFIDENTIAL EVIDENCE IN SUPPORT OF DEFENDANT ANDREW BUNNER'S MOTION FOR SUMMARY JUDGMENT

INDEX

Ex. 1

RICHARD R. WIEBE (SBN 121156)
425 California Street, Suite 2025
San Francisco, CA 94104
Telephone: (415) 433-3200
Facsimile: (415) 433-6382

THOMAS E. MOORE III (SBN 115107)
TOMLINSON ZISKO MOROSOLI & MASER LLP
200 Page Mill Road, Second Floor
Palo Alto, CA 94306
Telephone: (650) 325-8666
Facsimile: (650) 324-1808

ALLONN E. LEVY (SBN 187251)
HS LAW GROUP
210 N. Fourth St., Suite 201
San Jose, CA 95112
Telephone: (408) 295-7034
Facsimile: (408) 295-5799

CINDY A. COHN (SBN 145997)
ROBIN D. GROSS (SBN 200701)
ELECTRONIC FRONTIER FOUNDATION
454 Shotwell Street
San Francisco CA 94110
Telephone: (415) 436-9333
Facsimile: (415) 436-9993

Attorneys for Defendant ANDREW BUNNER

SUPERIOR COURT OF THE STATE OF CALIFORNIA

COUNTY OF SANTA CLARA

DVD COPY CONTROL ASSOCIATION, INC., Plaintiff, v. ANDREW THOMAS MCLAUGHLIN; ANDREW BUNNER; et al., Defendants.	Case No. CV - 786804 **DECLARATION OF PROFESSOR EDWARD W. FELTEN** **IN SUPPPORT OF DEFENDANT ANDREW BUNNER'S MOTION FOR SUMMARY JUDGMENT**

I, Professor Edward W. Felten, declare:

I. Introduction

1. My name is Edward W. Felten. I am a tenured Associate Professor of Computer Science at Princeton University, and I am Director of Princeton's Secure Internet Programming Laboratory. I received my Ph.D. in Computer Science and Engineering from the University of Washington in 1993, and my B.S. in Physics from the California Institute of Technology in 1985. I have been on the faculty at Princeton for about eight years.

2. For the 2001-2002 academic year, I am on sabbatical leave from Princeton, at the Center for Internet and Society at Stanford Law School. The Center focuses on interactions between technology and the law. I chose to spend my sabbatical year at the Center because of my increasing concern over the impact of new laws and court decisions on technologists. Cases like this one affect the environment in which legitimate computer security researchers and practitioners work. I myself have been restricted in my work by the Digital Millennium Copyright Act, as I describe in ¶ 11 below.

3. My main area of research and teaching is computer security, and my other research interests include operating systems, computer networks, and Internet software. I have published more than fifty papers in the research literature, and am the co-author of two books.

4. At Princeton I have created and taught courses on Information Security, Applied Cryptography, and Distributed Computing and Networking.

5. I have received a number of awards for my research, including a National Young Investigator award from the National Science Foundation, and an Alfred P. Sloan Foundation Fellowship. I have received Outstanding Paper or Best Paper awards at two conferences: in 1997 at the Symposium on Operating Systems Principles, the most prestigious academic conference on operating systems, and in 1995 at SIGMETRICS, the most prestigious conference on computer system performance analysis. I have given numerous special and invited talks at academic conferences.

6. I am the primary computer science expert witness for the U.S. Department of Justice in the ongoing antitrust case against Microsoft, *United States v. Microsoft*. In that capacity, I testified twice at trial and also filed a lengthy declaration in the remedy phase of that proceeding. I also advised the Justice Department extensively during the recently concluded settlement negotiations in that case.

7. I have also worked extensively with law enforcement agencies. I assisted the U.S. Attorney's office and the FBI with the "Melissa virus" case and a few other matters.

8. My research has been funded by government agencies, including the National Science Foundation and the Defense Advanced Research Projects Agency, and by industrial grants or gifts from IBM, Intel, Microsoft, Merrill Lynch, Sun Microsystems, Telcordia, and Trintech.

9. I have been appointed to advisory boards and study panels by industrial, professional, and governmental organizations. Sun Microsystems, Inc. appointed me to its Java Security Advisory Council, and I serve on Technical Advisory Boards for several other companies. The Institute for Defense Analyses[1], working in conjunction with the U.S. Department of Defense, chose me to serve in the Defense Science Study Group, and I obtained a U.S. "Secret" security clearance for that purpose. The Defense Advanced Research Projects Agency (DARPA), which is the main research arm of the Department of Defense, appointed me to its Information Science and Technology advisory board. The National Research Council (which consists of the National Academy of Sciences, the National Academy of Engineering, and the Institute of Medicine) appointed me to its study committee on "Fundamentals of Computer Science." The Association for Computing Machinery (ACM), which is the leading international professional society for computer

[1] The Institute for Defense Analyses is a nonprofit corporation whose purpose is to promote national security and the public interest and whose primary mission is to assist the Office of the Secretary of Defense, the Joint Chiefs of Staff, the unified military commands, and defense agencies in addressing important national security issues, particularly those requiring scientific and technical expertise.

scientists, appointed me to its Advisory Committee on Security and Privacy. I also serve as the moderator of the ACM Forum on Legal Regulation of Technology.

10. My research has been covered extensively in the national press. I have been quoted or profiled on numerous occasions in publications such as the New York Times, the Washington Post, the Wall Street Journal, and Newsweek.

11. I have been personally affected in my academic work by the uncertainty and restrictions generated by the application of new laws and court decisions to the field of computer science. Last year, I led a team of researchers who performed research on a set of proposed digital music copy protection schemes. On the eve of presenting and publishing our results on the significant flaws of these schemes at an academic conference, the Recording Industry Association of America (RIAA) and others threatened to sue us under the Digital Millennium Copyright Act (DCMA). They demanded the right to censor our research paper and our lecture. The chilling effect of this litigation threat caused us to initially withhold publication of our results and cancel the lecture rather than risk violating the DCMA. Because of the importance of ensuring the freedom of researchers to publish the results of their research, we brought a federal court action for declaratory relief. The RIAA and the other defendants subsequently stated they would not sue us under the DCMA, and so we published a paper and gave a lecture on our research at another scientific conference last August. The RIAA and the other defendants still claim a right to censor our further writing and speech on the topic, so our declaratory relief action is still pending.

II. CSS Is Not A Secret

12. I am familiar with the "Content Scrambling System" ("CSS") used to encrypt DVD movie disks. I understand that Plaintiff claims that the CSS algorithm and its keys remain a secret that is not generally known. As an active participant in the computer security research community, I can state with confidence that this claim is wrong.

13. It is wrong for at least two reasons. First, DeCSS has been, and continues to be, widely available from sources other than Mr. Bunner. Second, even independent of the

availability of DeCSS, both the CSS algorithm itself, and methods for determining the keys it uses, are now widely known.

III. DeCSS Continues To Be Widely Available

14. The source code for DeCSS is available at many places on the Internet. These can be found easily with a search engine. As I was writing this paragraph, I stopped to do a Web search for the term "DeCSS source code" on the Google search engine. It took me less than fifteen seconds to find a copy of DeCSS.

15. Some DeCSS Web sites are widely known and discussed. For example, Dr. David Touretzky at Carnegie Mellon University runs a site called "Gallery of CSS Descramblers," at http://www.cs.cmu.edu/~dst/DeCSS/Gallery, which contains code and procedures for descrambling CSS, expressed in many forms and media, including several computer languages. This site has been mentioned many times in court testimony and in the popular press. Its existence is common knowledge in the computer security research community.

16. In my everyday discussions with students, I have observed that many computer science students know what DeCSS is and know how to get it.

IV. The CSS Algorithm And The Keys It Uses Are Widely Known

17. Even independent of DeCSS, the details of the CSS algorithm are available on the Internet and are widely known. For example, a well-known paper by Frank Stevenson (entitled "Cryptanalysis of Contents Scrambling System") describes how CSS works and what its weaknesses are. This paper continues to be available on several Web sites. It can be found in seconds by doing a Web search on its title, or on its author's name. A search for the term "Frank Stevenson" on the Google search engine returns many links to Stevenson's paper, including one at http://www.cs.cmu.edu/~dst/DeCSS/FrankStevenson/analysis.html.

18. Stevenson's research is widely known and discussed in the computer security research community.

19. For example, not long after Stevenson's paper was published, I gave an informal seminar talk about it at Princeton. The audience was a room full of faculty, graduate

students, and undergraduates. I have also used DeCSS, CSS, and Stevenson's results as an example in one of the lectures of my senior-level Information Security course.

20. Although Stevenson's paper does not provide the CSS cryptographic keys, it describes methods by which those keys can be determined. These methods are well within the means and expertise of a typical computer science student, and do not require any rare tools: an ordinary personal computer and a few DVDs suffice.

21. As these facts demonstrate, neither CSS nor the keys it uses remain secret.

V. CSS and its Keys Would Inevitably Have Become Public

22. I understand that Plaintiff chose to allow wide distribution of DVD player computer software programs, running on personal computers, implementing CSS and containing valid keys. This decision to authorize software DVD players and not to limit DVD players to only hardware versions made it virtually inevitable that knowledge of CSS and its keys would become public.

23. Personal computer software is inherently amenable to reverse engineering. The tools to do this reverse engineering are widely available at little or no cost and run on ordinary personal computers. There are, at the very least, hundreds of thousands of people worldwide who have the skill to use them.

24. Reverse engineering tools for personal computer software are so good, and so widely available, because they have other valuable uses, especially in "debugging" software. Programmers spend many hours debugging the software they have written (i.e., diagnosing its malfunctions in order to fix them). Debugging is essentially the process of reverse-engineering your own software, so that you can figure out how its behavior differs from the behavior you desire. Any skilled programmer is good at debugging; and debugging is just reverse engineering. Applying the same tools, and many of the same methods, to software implementations of CSS, would yield an understanding of how CSS works.

25. Although some products exist that claim to "harden" software against reverse engineering, these products generally impair the performance of the "hardened" software,

and have only a limited practical effect against a skilled reverse engineer. Indeed, a recent discovery in theoretical computer science[2] proves that it is *impossible* to build a tool that effectively hardens arbitrary programs.

26. Because so many people have the skills and tools to reverse-engineer programs, Plaintiff's decision to authorize the release of CSS in software form made it virtually inevitable that somebody, somewhere, would reverse engineer it. It is hard to imagine that Plaintiff did not foresee this.

27. Once CSS became public knowledge, its keys inevitably also would have become public knowledge. This is true because the designers of CSS made the "rookie mistake" of using only a forty-bit key. It is common knowledge that use of a forty-bit key allows an easy brute-force search to determine the key, given a sample of encrypted material (e.g., a DVD movie disk). It is virtually impossible to imagine that Plaintiff did not realize this.

28. In fact, because the designers of CSS made the additional "rookie mistake" of using a home-grown cryptosystem rather than an "industrial-strength" one, it was not even necessary to search the entire 40-bit "key space" (i.e., the mathematical universe of all possible 40-bit numbers) to determine the working keys. Frank Stevenson was apparently the first to notice this, but the flaws in CSS were not terribly difficult to find. Finding the flaws in CSS would in fact make a good homework problem for a course in cryptography. It seems unlikely that Plaintiff could have done a due-diligence evaluation of CSS without learning of these additional flaws.

29. These facts demonstrate that Plaintiff's decision to allow personal computer software implementations of CSS made it virtually inevitable that CSS and its keys would become public knowledge. From my experience in the academic, commercial, government, and national security arenas of computer science, I know that this is not how businesses and individuals normally treat valuable information they desire to keep secret. In my view, the

[2] "On the (Im)possibility of Obfuscating Programs," by Barak, et al., Proceedings of the 21st International Conference on Cryptology, Santa Barbara, August 2001.

actions taken by Plaintiff cannot be considered reasonable efforts to maintain the secrecy of the CSS algorithm and keys.

I, EDWARD W. FELTEN, declare under penalty of perjury under the laws of the State of California that the foregoing is true and correct.

Dated: Nov. 15, 2001

Edward W. Felten

Ex. 2

RICHARD R. WIEBE (SBN 121156)
425 California Street, Suite 2025
San Francisco, CA 94104
Telephone: (415) 433-3200
Facsimile: (415) 433-6382

THOMAS E. MOORE III (SBN 115107)
TOMLINSON ZISKO MOROSOLI & MASER LLP
200 Page Mill Road, Second Floor
Palo Alto, CA 94306
Telephone: (650) 325-8666
Facsimile: (650) 324-1808

ALLONN E. LEVY (SBN 187251)
HS LAW GROUP
210 N. Fourth St. Second Floor
San Jose, CA 95112
Telephone: (408) 295-7034
Facsimile: (408) 295-5799

CINDY A. COHN (SBN 145997)
ROBIN D. GROSS (SBN 200701)
ELECTRONIC FRONTIER FOUNDATION
454 Shotwell Street
San Francisco CA 94110
Telephone: (415) 436-9333
Facsimile: (415) 436-9993

Attorneys for Defendant ANDREW BUNNER

SUPERIOR COURT OF THE STATE OF CALIFORNIA

COUNTY OF SANTA CLARA

DVD COPY CONTROL ASSOCIATION, INC., Plaintiff, v. ANDREW THOMAS MCLAUGHLIN; ANDREW BUNNER; et al., Defendants.	Case No. CV - 786804 **DECLARATION OF PROFESSOR DAVID A. WAGNER** **IN SUPPPORT OF DEFENDANT ANDREW BUNNER'S MOTION FOR SUMMARY JUDGMENT**

I, Professor David A. Wagner, declare:

1. I am an Assistant Professor of Computer Science at the University of California, Berkeley. I received an A.B. in Mathematics from Princeton University in 1995, a M.S. in Computer Science from Berkeley in 1999, and a Ph.D. in Computer Science from Berkeley in 2000. I am personally familiar with the facts set forth herein, and if called as a witness, I could and would testify them of my own personal knowledge.

2. My area of research includes computer and telecommunications security, cryptography, privacy, anonymity, and electronic commerce. Cryptography is the science of designing and analyzing secure codes and ciphers. I have published over 50 papers and 2 books on the subjects of cryptography and the security of computer systems. I also teach "Security in Computer Systems" at Berkeley, a graduate-level course on modern computer and network security.

3. My consulting work (I have done data security consulting through Counterpane Systems, Minneapolis, and independently), my studies (in addition to my work at Princeton and Berkeley, I twice interned at Bell Labs, studying under S. Bellovin) and my teaching and research have given me extensive experience in the analysis of real-world security systems. The systems I have personally examined include supposedly secure systems used by hundreds of millions of people. Many of my discoveries have resulted not only in academic publications, but also in widespread news coverage in leading newspapers, magazines, and TV news shows. For example, in September 1995, a colleague and I reported serious security flaws in the techniques used for encrypting credit card numbers in the leading products facilitating the implementation of electronic commerce over the Internet. This discovery was reported on the front page of the New York Times, the front page of the business section of the Washington Post, and elsewhere.

4. In March 1997, two colleagues and I reported on the flaws in the privacy codes used by U.S. digital cellular phones, phones used by tens of millions of U.S. citizens. This work not only received widespread news coverage (e.g., the front page of the New York Times), but also helped convince the U.S. cellular standard committee to undertake a sweeping redesign of their security architecture.

5. In April 1998, two colleagues and I reported on the weaknesses in the privacy and billing-security protections found in GSM digital cellular phones. GSM is the European cellular telephony standard, with over two hundred million users worldwide. Again, this work received widespread coverage in leading newspapers such as the front page of the business section of the New York Times, page A3 of the Wall Street Journal, and other similar publications.

DVD DECRYPTION

6. I have followed DVD security and encryption issues with interest, particularly after full details of the copy protection system were first publicly revealed in October 1999. The DVD copy protection system, which sometimes goes by the name "CSS," includes several components: the CSS cipher, the CSS authentication protocol, and the cryptographic keys associated with these algorithms. These are sometimes jointly referred to under the name CSS, but strictly speaking they are each distinct components.

7. A number of programs have been developed that allow users to view encrypted DVD movie disks. The DeCSS computer program was one of the first to achieve this by breaking the DVD encryption, but it is not the only one. DeCSS includes information that effectively discloses all three components of the CSS system (the CSS cipher, the CSS authentication protocol, and some of the cryptographic keys), but this information has been disclosed in other forms as well, as I discuss in detail below.

8. The term DeCSS has been used to refer to several DVD descrambling programs distributed in several different forms of computer code. Of relevance here are the binary executable code form of the program (commonly filenamed decss.exe), the source code for that binary version (which I shall refer to as decss-source), and the source code for a slightly different version of the program in the C programming language (commonly filenamed css-auth).

9. During the last week in October, 2001, two years after the first disclosure of the full details of CSS, I performed detailed experiments to assess whether full information on CSS remains accessible to the public on the Internet.

10. After careful examination, it is my conclusion that full information on the CSS technology is widely available on the Internet and elsewhere. I have verified that the relevant

information can be found in literally hundreds of places on the Internet. I will detail below the experimental methodology I used to come to this conclusion.

THE DECSS SOURCE CODE REMAINS WIDELY AVAILABLE AND REVEALS THE WORKINGS OF CSS

11. A URL is an address used to designate the location of a document on the Internet; with knowledge of the URL, anyone in the world can view that document. A good analogy is that a URL can be compared to a scholarly citation to a document, except that URLs are specially designed for referring to documents available over the Internet.

12. I began with a list of 465 Internet URLs to determine whether any of the documents those URLs identified contain information on CSS.

13. I know that the Internet changes rapidly, and that documents on the Internet sometime become unavailable over time, for instance if the publisher of the document changes addresses. Therefore, as a first step, I visited each of the 465 documents referred to by these URLs to verify which ones remain accessible on the Internet. I was unable to view 49 of these documents, but I verified that the remaining 416 of these were accessible on the Internet to me.

14. Sometimes two different URLs can refer to the same document at the same location: they might refer to two slightly different pathways to access the same location. (You can imagine that there might be two ways of writing a citation to the same document, according to, for instance, how the title is capitalized in the citation. This gives a good analogy for what I am talking about here.) I screened the list for various ways that this could happen, and of these 416 URLs, 22 appeared to be duplicates. I made a copy of each of the remaining 394 documents.

15. Next, I manually examined these 394 documents to identify which ones disclose information about CSS. Many of these documents were copies of each other, made available from different locations, and this made my identification task somewhat easier. I classified the documents according to what information they revealed.

16. I found that 1 of these 394 documents contained essentially no information about CSS, so I discarded it from further analysis.

17. I found that 10 more of these documents disclosed information about only one component of the CSS system: they appeared to be lists of cryptographic keys (specifically, player keys) used by CSS.

18. Of the remaining 383 documents, I found that 164 contained DeCSS in binary executable form only-the decss.exe program. As mentioned above DeCSS is available both as a binary which can be executed on a computer (decss.exe) and in two different source code versions which can be easily read by a programmer (decss-source and css-auth). In binary form (binary code is sometimes also referred to object code), DeCSS does contain very detailed information about all three components of CSS, and this information could be extracted by a dedicated programmer, but not easily (it would likely require hours of work). In contrast, the source code versions are designed to be easily understood by programmers and thus reveal detailed information about CSS in a very clear and explicit form. Thus, these 164 DeCSS binary program documents can be viewed as revealing much information about CSS, but in a form that requires some work for a human to read. This was the only category of documents that was not easily readable with the naked eye.

19. All of the remaining 219 documents contained information about CSS in source code version, either css-auth or decss-source. This source code is easily readable by people trained in computer programming. The source code available at these 219 sites contained very detailed information about all three components of CSS, including a full specification of the CSS cipher, the authentication protocol, and some of the cryptographic keys. Each of these documents contained enough information to reveal essentially everything about CSS and how it operates to descramble a DVD movie disk.

20. This shows that CSS is currently available in an easily understandable source code form from hundreds of places (at a minimum) on the Internet. (Again, this excludes the 164 additional sites from which I found the binary executable form of DeCSS to be available.)

21. At this point, I would like to inject a few words of caution about how to interpret this conclusion. Documents on the Internet come and go. Documents are not perpetually archived, but remain available only so long as their publisher makes them so, and new

documents are added and deleted frequently. Because of this constant churn, any experiment can only reveal what is accessible at the time the experiment was performed, and results might vary if the experiment is repeated later. Moreover, I should warn that because I began with a limited list of 465 Internet URLs (only a tiny fraction of the 1.6 billion web pages indexed by the Google web search service, for example), my experiment might greatly under-estimate the number of places where CSS can be found on the Internet. My experiment shows that CSS source code is available from at least 219 sites on the Internet, but it is entirely possible that the true number might be larger by a factor of 10 or more. For example, using the Google web search service to search for the term "decss source code" returned about 10,400 hits, and a Google search for the term "css auth" returned about 15,600 hits.

OTHER DVD DECRAMBLING PROGRAMS ARE ALSO WIDELY AVAILABLE AND REVEAL THE WORKINGS OF CSS

22. I next performed a second experiment to assess the availability of information on CSS from other sources. Because any DVD player that can display encrypted DVD's must contain the CSS descrambling technology, I hypothesized that other open-source DVD players might also reveal similar information about CSS. I used the Google web search service to find other open-source DVD players.

23. After a few hours of searching, I found 11 other source code software packages that disclosed very detailed information about CSS. These 11 were the DVD players known by the following names: DeCSSplus, DecVOB, DVDPlayer, Livid, Ogle, VideoLAN, VobDec+, vStrip, xine_d4d_plugin, complete_xine, and xine_css_dvd. Each of these software packages were readily available to the public in source code form and seemed to my inspection to reveal essentially full information about CSS.

24. I did not try to assess how many places these software packages might be available from. It is possible that each of these 11 software packages is available from only one place. It is also possible that, like DeCSS, many of these packages are available from many different places on the Internet. I did not try to check. I stress, however, that these software

packages can be easily found by any computer-literate person who wishes to find them; I did not use any special techniques or services to locate this information.

25. In summary, the second experiment supports the conclusion that detailed information about CSS is disclosed not only by DeCSS but also by a good deal of other DVD descrambling software widely available on the Internet.

OTHER SOURCES OF INFORMATION ABOUT CSS ARE ALSO WIDELY AVAILABLE

26. Next, I performed a third experiment. I knew that Exhibit B of the reply declaration of John J. Hoy dated January 18, 2000 (the "Hoy reply") revealed very detailed information about CSS, including the CSS cipher and a CSS player key. Again using the Google search service, I immediately found 6 places on the Internet where exact copies of Exhibit B of the Hoy reply could be obtained, including at least two different academic web sites: a publicly-accessible Harvard University web site at http://eon.law.harvard.edu/openlaw/DVD/resources/dvd-hoy-reply.html and a publicly accessible Case-Western Reserve University web site at http://samsara.law.cwru.edu/dmca/csscode.html. (In the process, I encountered a number of other documents that also revealed as much or more information on CSS as Exhibit B of the Hoy reply did, but they were not exact copies of Exhibit B, so I ignored them.) I conclude that the CSS information contained in Exhibit B of the Hoy reply is readily available to all interested parties.

27. In light of these experiments, I conclude all relevant technical information on CSS is readily available to the public.

THE FAILINGS OF CSS HAVE BECOME A COMPUTER SCIENCE AND CRYPTOGRAPHY TEACHING TOOL

28. I have used this publicly-available information about the CSS system in my teaching. When I last taught my graduate course on "Security in Computer Systems," I gave one lecture on the topic of copy protection and DVD security. As usual. I consulted a number of primary and secondary sources in preparing this lecture, and for this lecture these sources included the October 1999 Internet discussions about CSS, Frank Stevenson's paper analyzing

the cryptographic properties of CSS, various documents written by the designers of the DVD security architecture, the DeCSS computer program, scholarly analysis of information about CSS by several researchers, and a number of other documents available on the Internet, including the Hoy reply. In my lecture, I presented the CSS DVD security system as an example of a failed security system where students could learn from the designer's mistakes. The publicly-available information on CSS I found enabled me to give specific details that helped students to better understand the design choices made in CSS and the reasons why CSS failed as a security system. I believe being able to give concrete, specific details on real-world security systems and their vulnerabilities and failures helps students learn more effectively than they could in any other way.

29. The flaws of CSS that make it a useful example for academic teaching and discussion led to its failure as a real-world security system. I believe that any competent cryptographer with full knowledge of the design of the DVD security system would have expressed serious reservations about the ability of the system to withstand scrutiny. The cipher was a weak one, within the abilities of a graduate-level cryptography student to break with an ordinary PC. CSS also relied on distributing software in an "obscured" form -- hidden in locations that are not immediately obvious. Many manufacturers distribute security systems in an obscured form in the hopes that no one will bother to take the time to reverse engineer their inner workings. In my opinion, this is a foolish and immature judgment: when one's system is distributed to millions of individuals around the world, it is imprudent to assume that no one will take an interest in the system's operation. From a security point of view, attempting to keep the inner workings of your security system secret merely by concealing its parts is ultimately futile and serves little purpose.

30. Information about the cryptographic flaws in CSS was widely distributed within the academic research community, and to other cryptographers (many of whom do important work although they lack any academic or institutional affiliation), over the Internet at the time DeCSS was first released in October 1999. The flaws in DVD security were a topic of extensive discussion and continue to be widely known within the cryptographic community.

31. Investigation and publication of these types of flaws in supposedly secure systems serves a vital public interest. As our society becomes increasingly dependent on computers, telecommunications, and other information systems, it is important that these systems be trustworthy and free of systemic security flaws. For example, as electronic commerce becomes more prevalent, criminals gain an increasing financial incentive to exploit security vulnerabilities in those systems. The cellular phone and electronic commerce security vulnerabilities I have investigated and described above clearly illustrate that the risks are very real: much of our existing infrastructure contains serious security vulnerabilities in its design and implementation, even though this fact may not be widely known to the public. I believe that it is the scientific community's duty to study these issues and to report on security vulnerabilities that the public at large may not be aware of. One must understand the vulnerabilities and flaws of existing security systems in order to prevent them from recurring.

32. Progress in the sciences of cryptography and computer security is dependent on investigation of existing, widely-used security systems and public disclosure of whatever flaws are found. It is widely understood in the cryptographic community that the only way to learn how to build secure systems is to be intimately aware of the techniques a typical attacker might use: to be a good codemaker, one must be an accomplished code breaker. Moreover, it is not enough merely to study the theory of code-breaking: it is crucial to understand how real-world security measures are broken in practice if we wish to build and deploy real security systems that are highly resistant to attack.

33. Publication and circulation of results of security system investigations is the accepted and necessary method for sharing ideas and advancing scientific knowledge about cryptography, just as in every other science. The combined knowledge of the cryptography research community is defined by published results, and extending the body of knowledge on how real-world systems get broken in practice is crucial to securing the systems of the future. Those who do not know history are condemned to repeat it; and publication is how the cryptography community comes to know the history of what has succeeded and failed in the past.

THE WORKINGS OF CSS ARE WIDELY KNOWN BECAUSE OF DECISIONS MADE BY THOSE WHO DESIGNED AND IMPLEMENTED CSS

34. The cryptographic flaws of CSS discussed above, including its weak cipher, its choice of a 40-bit key length and its failure to maximize the cryptographic strength of its 40-bit keys, and its reliance on obscurity as a security technique, were not the only factors that led to the widespread public knowledge of the CSS algorithms and keys.

35. Perhaps the most significant factor in the reverse engineering and public knowledge of CSS was the choice of the creators and licensors of CSS to permit it to be implemented in authorized DVD software players. Once they decided to permit software versions of CSS, it was inevitable that the CSS algorithms and keys would become public knowledge in a relatively short time. Moreover, because each software implementation contains essentially full information on CSS, once a single software implementation is reverse engineered, all the details are revealed.

36. It is widely understood in the cryptographic community that software implementations of computer security systems are much less resistant to reverse engineering than are hardware implementations of the same systems. Hardware implementations, in which the desired computer operations are hardwired into the circuitry of a special-purpose microprocessor, are more resistant because reverse engineering them requires skills, techniques, and machines that are uncommon. For example, the security system used in Europe's GSM mobile phones remained secure for over 10 years, despite being used by hundreds of millions of users, because it was implemented in hardware. A given system implemented in tamper-resistant hardware might have a typical lifetime of 5 to 15 years before being reverse engineered; the same implementation in ordinary hardware might have a lifetime of 5 to 10 years; the same implementation in software might have a lifetime of only 2 to 3 years before being reverse engineered.

37. There are several reasons why software security systems are much more vulnerable than hardware systems. First, the human skills and the machines necessary to reverse engineer software are much more common and much less specialized than those required to

reverse engineer hardware. Software can often be reverse engineered with only an ordinary PC and a basic understanding of computer programming.

38. Second, software is inherently subject to reverse engineering in a way that hardware is not because, in order to control the operations of a computer, the software must be translated into an electrical signal that travels within the computer from the software storage device to the central processing unit. This electrical signal may be observed and decoded to reveal the message of the software. Moreover, observation is usually possible with standard software tools: one can use one piece of software to observe what another piece of software is doing.

39. Thus, with software security systems, it is only a matter of time, usually a short time, before someone with the skills and the interest to reverse engineer it comes along.

40. For these reasons, cryptographers understand that implementing a security system in software does not provide a reasonable level of precaution against public disclosure. No software implementation of a data copy protection scheme that I know of has ever successfully resisted reverse engineering for long. Just recently, for example, the digital rights management scheme used to protect Windows ".wma" format audio files was broken and publicly revealed. This was actually the second time the copy protection on ".wma" files was broken: on August 18th, 1999, a free utility was released that broke an earlier version of the copy protection scheme—just one day after that copy protection scheme was officially released.

I, DAVID A. WAGNER , declare under penalty of perjury under the laws of the State of California that the foregoing is true and correct.

Dated: 11/15/2001

David A. Wagner

Ex. 3

RICHARD R. WIEBE (SBN 121156)
425 California Street, Suite 2025
San Francisco, CA 94104
Telephone: (415) 433-3200
Facsimile: (415) 433-6382

THOMAS E. MOORE III (SBN 115107)
TOMLINSON ZISKO MOROSOLI & MASER LLP
200 Page Mill Road, Second Floor
Palo Alto, CA 94306
Telephone: (650) 325-8666
Facsimile: (650) 324-1808

ALLONN E. LEVY (SBN 187251)
HS LAW GROUP
210 N. Fourth St., Suite 201
San Jose, CA 95112
Telephone: (408) 295-7034
Facsimile: (408) 295-5799

ROBIN D. GROSS (SBN 200701)
ELECTRONIC FRONTIER FOUNDATION
454 Shotwell Street
San Francisco CA 94110
Telephone: (415) 436-9333
Facsimile: (415) 436-9993

Attorneys for Defendant ANDREW BUNNER

SUPERIOR COURT OF THE STATE OF CALIFORNIA

COUNTY OF SANTA CLARA

DVD COPY CONTROL ASSOCIATION, INC., Plaintiff, v. ANDREW THOMAS MCLAUGHLIN; ANDREW BUNNER; et al., Defendants.	Case No. CV - 786804 **DECLARATION OF COMPUTER SCIENTIST DAVID S. TOURETZKY** **IN SUPPPORT OF DEFENDANT ANDREW BUNNER'S MOTION FOR SUMMARY JUDGMENT**

I, DAVID S. TOURETZKY, declare:

1. I am currently a Principal Scientist in the Computer Science Department and the Center for the Neural Basis of Cognition at Carnegie Mellon University, in Pittsburgh, Pennsylvania. I earned both my M.S. and Ph.D. degrees in Computer Science from Carnegie Mellon University. I lecture regularly around the world on such topics as cognitive science, artificial intelligence, robotics, and neural networks. I have authored three books, edited or co-edited nine collections of scholarly works, and authored or co-authored dozens of articles for scholarly journals, conference presentations, and the like. Over the past 25 years I have taught computer science material in a variety of formats, including brief tutorials at national conferences, week-long seminars for industrial clients, and semester-length university courses.

2. I have been interested in the issues surrounding DVD encryption since first hearing about this case in December 1999. At that time, I learned of two DVD decryption programs. The first is DECSS.EXE, a decryption program written for the Microsoft Windows family of operating systems. The second, known as css-auth, is written for Linux, a version of the Unix operating system. Both programs allow users to access a DVD drive and decrypt a DVD movie. The term "DeCSS" originally referred to DECSS.EXE, but has since been used as a generic term for any piece of software that defeats CSS encryption. Therefore, in this declaration I will avoid using "DeCSS" and instead refer explicitly to various DVD decryption programs by name (e.g., DECSS.EXE or css-auth).

EXPLANATION OF CSS ENCRYPTION TECHNOLOGY

3. The sounds and images of movies are translated into digital form for storage and playback by computers and other electronic devices. The information is stored in a publicly-disclosed file format called MPEG, which contains no encryption or access limitation technology. Software for recording and playing MPEG files is widely available.

4. In order to control access to the content distributed on DVD movie disks, motion picture studios encrypt their MPEG movie audiovisual data using a scheme called CSS (Content Scrambling System). The CSS-encrypted MPEG movie data is divided into numerous separate files when it is stored on a DVD disk.

5. CSS is based on a type of encryption algorithm known as a "stream cipher," in which a message is encrypted by combining it mathematically with a stream of seemingly random bits (ones and zeros).. The stream is generated by a mathematical formula, or algorithm, based on a numerical password called a "key." The stream is not truly random because the algorithm will always produce the same result when given the same key as input; this is what allows the message to be decrypted later. CSS uses a 5 byte key (or equivalently, a 40 bit key, since a byte is a group of eight bits.) To recover the original message from a stream of encrypted bytes, one merely needs to know the 5 byte key that was used to initialize the stream generator; one can then recreate the stream of pseudo-random bits and subtract them from the encrypted data to obtain the unencrypted message.

6. When encrypted movies are distributed on DVDs, the disk must also contain the 5-byte key used to encrypt the movie data, so that the movie can be decrypted and viewed. The protection afforded by CSS is based on the assumptions that (a) consumers don't know how the files are encrypted, and (b) untrusted software running on the consumer's computer will not be able to get at the key, while an authorized DVD player program can. To achieve this, several measures are taken. First, before a program is allowed to access the data on a DVD drive, the DVD player program must "unlock" the drive by going through an authentication sequence with it. This authentication sequence involves an exchange of encrypted messages between the computer and the drive, using one of a set of 32 initial keys. In this way, the DVD player program "proves" to the drive that it knows the secret encryption scheme, and therefore is authorized to access the movie data on the disk.

7. This protection scheme is imperfect. One way around it is to use authorized software to unlock the drive and then switch over to unauthorized software. The drive cannot tell if the computer subsequently switches to a different, unauthorized program; it will continue to honor requests to access movie data files on the disk. Another problem is that the authentication sequence, including the set of 32 initial keys, has become widely known. Code to perform authentication is included as part of the css-auth package (in the file tstdvd.c), and is also included in various DVD player programs for Unix, such as Videolan (from the Ecole Centrale

Paris), Ogle (from Chalmers University of Technology in Sweden), and Xine. These players are "open source" programs, meaning their source code is freely distributed. (The Xine player requires a separate plug-in to unlock and decrypt a DVD. Source code for two different plug-ins with this functionality are available from third parties at the time of this writing.) Anyone interested can learn how to do DVD drive authentication by spending a few minutes reading some of this code. I recommend Videolan's vlc-dvd_css.c file.

8. CSS includes another way to protect DVD content even if the drive is unlocked. The key used to encrypt each movie file (called a "title key") is itself encrypted using a "disk key" that is unique to that disk. And the disk key is itself encrypted using each of 409 "master keys." Given any valid master key, one can decrypt the disk key, then use the disk key to decrypt each title key, and then use the title keys to decrypt the movie. Master keys were kept secret in an attempt to prevent this.

9. As a further precaution, when the disk and title keys are sent to the DVD player program by the DVD drive, they are encrypted using a "session key" exchanged between the drive and the DVD player program as part of the initial authentication process. This prevents the capture of unencrypted disk and title keys by eavesdropping on the computer's input/output bus.

10. Master keys (also called player keys) are not stored on the disk; they are stored either in a chip on a circuit board (in the case of a hardware DVD player) or embedded in an obscured fashion in a piece of executable software (in the case of software DVD players). Different DVD hardware and software player products were assigned different player keys so that if a particular player key were to be disclosed, the studios could simply stop using that key in any future DVD releases. This has in fact already happened. The Xing software DVD player's master key was revealed in 1999. The studios then discontinued use of this key, so players that rely on it are unable to play new movies. Both DECSS.EXE and css-auth employ the Xing key. The key has also been published in the Wall Street Journal, in haiku form ("Banned Code Lives in Poetry and Song", by David P. Hamilton, April 12, 2001, page B1, a copy of which is attached as Exhibit A).

11. More recent DVD decryption programs, such as VobDec, do not rely on player keys. They obtain the title key directly through a type of mathematical analysis known as a cryptographic attack. This is possible because the CSS stream cipher was poorly designed, as documented by Frank Stevenson. Mr. Stevenson's research paper on this topic, entitled "Cryptanalysis of Contents Scrambling System," has been widely circulated on the web, and is archived as part of my Gallery of CSS Descramblers web site, discussed below.

12. What Mr. Stevenson showed was that the mathematical function CSS uses to generate a stream of pseudo-random bits has certain predictable qualities, and as a result, one can make educated guesses about the title key that was used to encrypt a particular sequence of bytes, then test each guess. Due to a flaw in the design, the number of tests required to discover the title key is far less than it should be. In fact, it is small enough that a modern computer can uncover the title key in less than a minute. Mr. Stevenson also showed how the weaknesses in the encryption of the disk key could be used to recover all the player keys, and this was done in 1999. (See the www.free-dvd.org.lu web site, and the file www.free-dvd.org.lu/random-numbers.txt. The file name is an attempt at humor; the numbers are not random.) But as explained earlier, player keys are no longer needed now that the title key cipher's weaknesses are well understood.

THE CSS-AUTH SOURCE CODE HAS BEEN CONTINUOUSLY AVAILABLE SINCE THE BEGINNING OF THIS LITIGATION AND REMAINS WIDELY AVAILABLE

13. In December 1999 I established a "mirror" (local copy) of one of the DVD decryption programs, css-auth.tar.gz, on my web site at Carnegie Mellon. The css-auth.tar.gz file contains the software package css-auth. This mirror has remained continuously available on my web site from late December 1999 through today.

14. In March of 2000 I created a web site called the Gallery of CSS Descramblers, at http://www.cs.cmu.edu/~dst/DeCSS/Gallery (incorporated by reference in this declaration). I created this web site as a scholarly publication to illustrate the many forms an algorithm description could take, both in computer code and other forms of speech. My Gallery of CSS Descramblers presented a variety of exhibits, including the original css-auth source code in the C

programming language, a version of the css-auth code translated into a made-up computer language for which there was not yet a compiler (so, technically, it might not even be "code"), and a version of the css-auth code translated line-by-line into plain English.

15. The Gallery of CSS Descramblers has received extensive publicity and media coverage. On July 25, 2000, I testified as an expert witness for the defense in *Universal City Studios, et al. v. Reimerdes, et al.*, 111 F.Supp.2d 294 (S.D.N.Y 2000), commonly known as "the 2600 case." My testimony, which focused on the Gallery and the equivalence of computer code and other forms of speech, was featured in articles in the New York Times, the AP News wire service, the Hollywood Reporter, and several other publications. As a result, people began sending me contributions to the Gallery, in the form of computer code, audio recordings, graphic images, and animations. Each contribution expressed the css-auth source code or the underlying decryption algorithm in a creative way. For example, one person set the English description of the algorithm to music and sang it, with guitar and drum accompaniment. Another sent an image file in which the C program was cleverly encoded as a picture of Jack Valenti, president of the Motion Picture Association of America. And another person sent a 456-stanza haiku that included a complete and technically correct description of the css-auth decryption algorithm in perfect 5-7-5 syllable form.

16. The various exhibits added to the Gallery have resulted in additional media coverage, including articles in the New York Times, the Wall Street Journal, the Washington Post, the San Francisco Chronicle, Le Monde, the Bangkok Post, and Neue Zurcher Zeitung. USA Today named the Gallery a "Hot Site of the Day" for September 21, 2000. The Gallery now includes a collection of some 60 "press clippings," in the form of links to articles that discuss the Gallery or my testimony at trial. I have also made two television appearances to discuss the Gallery and the 2600 case. One was an interview on Tech TV's "Screen Savers;" the other was as a guest on John Dvorak's program, "Silicon Spin."

17. The Gallery has evolved to include not just representations of the css-auth code, but also technical descriptions and lecture notes about the CSS protection scheme and the decryption

algorithm, legal documents relating to the 2600 case, and links to web sites where other DVD decryption software can be found.

18. The Gallery is widely known on the Internet. Google, a popular Internet search engine (www.google.com), ranks its search results, or "hits," by the number of other sites that link to the site found by the search engine. A search for "DeCSS" using the Google Internet search engine on September 14, 2001 brought up the Gallery as the #2 hit out of a total of 77,800 hits returned. A reverse search from Google showed 594 sites with links to the Gallery, including links from Wired Magazine, USA Today, Slashdot, The Register, and the Association for Computing Machinery (the major professional organization for computer scientists.) The Gallery was also the first item listed in Google's human-edited directory on the topic "DVD CSS," which is part of the Cryptography section. See http://directory.google.com/Top/Society/Issues/Human_Rights_and_Liberties/Privacy/Cryptography/DVD_CSS.

19. DVD decryption software remains available from many other sources as well. On September 8, 2001, I used Google to performed a search for the string "css-auth.tar.gz." This is the name usually used for the file containing the source code of the css-auth package. The ".tar" extension denotes Tape ARchive format, which is a Unix convention for encapsulating a collection of files into one large file; the ".gz" extension indicates that the tar file has been compressed with a utility called gzip.

20. My search returned 830 hits, of which Google's heuristics decided 399 were likely to be unique pages. I examined the first 20 of these by visiting each link. There were 18 unique web sites in the first 20 hits. (Two sites were repeated due to hits on two separate pages on the same site.) Of those 18 unique sites, 9 contained local copies of css-auth.tar.gz, which I verified by downloading the file and either unzipping it or checking the file length in bytes. These sites were located in Austria, Denmark, Norway, the United Kingdom, and the United States. One was my own Gallery of CSS Descramblers. Another 8 of the 18 sites did not contain usable local copies of the file, but had links to other mirror sites where css-auth.tar.gz could be found. I followed some of those links and found additional copies of css-auth.tar.gz in Germany,

Luxembourg, the Netherlands, the United Kingdom, and the United States. The 18th site was down, but by retrieving a copy of the page from the Google cache I was able to determine that it was also a list of mirrors.

21. As a further test, I examined hits number 101 through 110 from the 399 results returned by Google. Each of these hits was a unique site, and none were included in the previous 20 results. 6 of these 10 sites contained local copies of css-auth.tar.gz; the servers were located in Germany, Switzerland, and the United States. Another site had a list of links to mirrors. Two of the sites were down. The tenth site, located in North Carolina, contained a press release and a link to the previously-mentioned Luxembourg site where the file could be found.

22. I also explored hits further down the list and found copies of css-auth.tar.gz on servers in Australia, France, Finland, New Zealand, and Poland.

23. Based on this experiment, I conclude that the css-auth source code remains widely available on the Internet, and can be found in a matter of seconds by anyone who bothers to look for it.

AVAILABILITY OF OTHER UNAUTHORIZED DVD SOFTWARE

24. Unauthorized DVD software falls into several categories: (1) Programs that capture individual frames from the computer's video card while the movie is being played by an authorized player. These were the first programs used to "rip" (capture and store) DVD movies, predating both DECSS.EXE and css-auth. They rely on an authorized player to do the actual decryption; they then intercept the movie's audiovisual data after it has been decrypted. (2) Programs that decrypt DVD movies and store them on the computer's hard drive. DECSS.EXE was the first decryption program in this category. The css-auth package also contains a program (css-cat.c) to do this. Many others have since been released, such as SmartRipper, VobDec, cladDVD, and DVD Decrypter. Some programs also compress the movie using a tool called DivX. Compression reduces the amount of disk space the movie takes up. (3) Programs that not only decrypt the movie but also play it on the computer's monitor and speakers, rather than storing it on the hard drive. Examples include LiViD (available at www.au.linuxvideo.org), Videolan (available at www.videolan.org), Ogle (available at

http://www.dtek.chalmers.se/groups/dvd), and Xine (available at xine.sourceforge.net). (4) Software packages that simply provide drive authentication and/or decryption services. These are components for use in constructing other programs. One example is the css-auth package previously discussed. Another is my Gallery of CSS Descramblers, which contains numerous implementations of the basic decryption algorithm.

25. There are many web sites devoted to the subject of DVD decryption software. Examples include www.flexion.org, www.doom9.net, and www.afterdawn.com, which are all located outside the United States. In addition to offering downloadable copies of the software itself, these pages include tutorials on DVD decryption and reviews of the strengths and weaknesses of different tools.

CSS AND THE COMPUTER SCIENCE ACADEMIC COMMUNITY

26. CSS is of interest to computer scientists for a number of reasons. It's one of the first examples of encryption technology embedded in a home entertainment product. It's also a stellar example of the failure of what experts call the "security through obscurity" approach. "Security through obscurity" refers to concealment of information about how a security mechanism works in the hopes that no attacker will uncover its weaknesses. The alternative is to develop mathematically strong encryption algorithms, publicly disclose them, and allow them to be examined by experts to determine if the algorithms are truly sound. CSS was not designed to withstand such scrutiny. CSS does not provide true security because the scheme is vulnerable to reverse engineering, the stream cipher is much weaker than theoretically possible due to flaws in its design, and in any case, the decryption keys must be present on each DVD sold. So CSS is an object lesson in how not to design a security product.

27. The application of the Digital Millennium Copyright Act to DECSS.EXE and css-auth in the 2600 case has raised the issue of the First Amendment status of computer code, a topic of vital concern to computer scientists and engineers. It has thus generated widespread interest in CSS decryption software among computer scientists and academics, even those, such as myself, who have no desire to watch DVD movies.

28. Here are some examples of how CSS has made its way into the computer science curriculum. Gregory Kesden, who teaches an undergraduate computer science course on Operating Systems at Carnegie Mellon University, now includes a lecture on the CSS encryption scheme. His lecture notes are available on the web at http://www.cs.cmu.edu/~dst/DeCSS/Kesden/index.html. Professor Greg Newby at the University of North Carolina also covers CSS in his course Distributed Systems and Analysis; see http://www.ils.unc.edu/gbnewby/DVD for more information.

29. MIT held a two-session seminar on "Decrypting DVD" in January 2001. The speakers included two undergraduates, Keith Winstein and Marc Horowitz, plus Professor Hal Abelson of the MIT Laboratory for Computer Science, Harvard Law School Professor Jonathan Zittrain of the Berkman Center for Internet & Society at Harvard Law School, and David Barr, lead engineer for C-Cubed Microsystems. As part of this event, Winstein and Horowitz dissected the CSS encryption scheme and presented the world's shortest CSS decryption algorithm: a 7-line program in the Perl computer language (later shortened to 6 lines). They demonstrated the algorithm's correctness for the audience by decrypting and playing a portion of the movie *The Matrix*. Their Perl program has been published in the July/August 2001 issue of the MIT-published journal Technology Review as part of the article "The Net Effect: The DVD Rebellion," by Simson Garfinkel. Technology Review is a print journal, but the article is also available on the web at http://www.technologyreview.com/magazine/jul01/garfinkel.asp. (A copy of the print version of this article is attached as Exhibit B.) Wired Magazine also published the source code in an article on March 7, 2001, available on the web at http://www.wired.com/news/culture/0,1284,42259,00.html. (A copy of the print version of this article is attached as Exhibit C.) The publication of the Winstein and Horowitz work inspired an MIT alumnus, Charles M. Hannum, to devise a 7-line C program to implement the same algorithm. Both these programs attracted considerable media attention, including a March 8, 2001 article in ZDNet News that was picked up by USA Today and MSNBC, plus articles in Slashdot and The Register. Further publicity came when Phil Carmody, a computer scientist in the United Kingdom, found ways to encode these tiny programs as prime numbers. More

information on these programs and their prime number encodings is available at the Gallery of CSS Descramblers.

30. Another indication of the growing familiarity with CSS in the computer science community is the appearance of new video playing software that includes DVD decryption. Videolan (www.videolan.org) was created as an academic project by a group at the Ecole Centrale Paris. A listing of the students involved and their faculty advisors may be found at http://www.videolan.org/team.html. Similarly, Ogle was created by a group at Chalmers University of Technology in Sweden; see http://www.dtek.chalmers.se/groups/dvd/authors.html for their names. Both Videolan and Ogle are distributed under the GNU Public License, allowing anyone to download and read the source code.

SUMMARY AND CONCLUSION

31. The technical details of how CSS works and how it can be defeated are now widely known. Not only are the early decryption programs DECSS.EXE and css-auth still available, but they have been joined by more sophisticated solutions using a cryptographic attack (based on Frank Stevenson's work), and a profusion of more refined DVD descrambling software that is both more reliable and easier to use. The story of how CSS was defeated will almost certainly be included in the next generation of security and cryptography textbooks as a perfect example of why the "security through obscurity" approach does not work.

32. At this point, there is nothing secret about DVD encryption. The cat has been long out of the bag. In fact, she's produced several litters of kittens.

I, DAVID S. TOURETZKY, declare under penalty of perjury under the laws of the State of California that the foregoing is true and correct.

Dated: Nov. 5, 2001

David S Touretzky

David S. Touretzky

Ex. A to Ex. 3

Wall Street Journal April 12, 2001
p. B1

...es: the automo-
...rs under the
...ger compartment,
analog island in
...tech digital

...t's about to
. The auto compa-
...e big auto-parts
...rs, and some
...er and electron-
...panies have all
...led that drivers
...empting market
...ital products and
...s. The trick is to
way to embed
...echnologies in
...ithout distracting
...ver and threaten-
...ety.
...a recent visit to
test-driving a
...tted with a baby
...player that could
...home-made CDs
...sonal computer.
...is that they can
...ecent quality, so
...Ds while flying

...d auto-parts com-
...e array of Inter-
...eloped to go into
...rs.
Lexus and BMW,
...al features for
...-free digital cell-
...at receive data
...as offered a
...n gives personal-
... luxury models.
and failed, to
...t called AutoPC.
...ends to push
...ers into moder-
that the MP3-CD
...on, the huge
...Motor last year.
...n lineup of digi-
...d GM's OnStar is
...xtra-cost service
...stock prices and
...omputerized
...ese automotive
...n time to time.
...s new Mach
...not great. The
...FM radio, is de-
...ir current car,
Visteon says
...st getting an
...less you're re-
...cars with large
...purchased on Vis-
...n.
...ny homemade
...and it handled
...around Detroit's
...risingly poorly
...stump the unit.
...MP3 produced a
...ng from track to

...numerous songs
...l the songs are
...rectory" mode,
...to any computer
...on has even
...vou quickly

Squibb Co. to relinquish patent rights for an AIDS drug in South Africa. And student activists at other

comes from the sheer scope of the AIDS epidemic in Africa, where 25 million people are infected

Swarr.

Ms. Swarr, whose field is women's studies, is a

application. "It's just sitting

Please Turn to Page B

Banned Code Lives in Poetry and Song

Critics of DVD-Copyright Ruling Say the Constitution Protects Posting Program in All Forms

By David P. Hamilton
Staff Reporter of The Wall Street Journal

IF DISSEMINATING a software code violates federal copyright law, does the First Amendment allow that code to be published in a poem or sung in a song? Critics of a sweeping court ruling may soon find out.

Ever since the software code, known as DeCSS, first showed up on the Internet about two years ago, Hollywood has feared that the Napsterization of movies was at hand. After all, the software program can break the encryption code protecting DVD movies from being copied. To block online sharing of DeCSS, the movie industry sued. Last August, U.S. District Judge Lewis Kaplan, citing federal copyright law, banned a hacker Web site called 2600.com from posting DeCSS or even linking to other sites that posted it.

But programmers and civil libertarians argue that software code deserves the same First Amendment protection as any other work of art. To underscore that point, a world-wide cabal of protesters is turning the code into a variety of artistic works that, they say, are indistinguishable from any other protected expression.

In February, for instance, a computer programmer lodged his complaint against Hollywood and the legal system with an epic poem—456 stanzas of haiku, no less—that makes an impassioned argument for the free expression of mathematical ideas. The poem, replete with references to Homeric epics and the history of mathematics, also contains a detailed explanation of the DeCSS algorithm.

"A program is a literary work," says the work's author. "The idea was to show how strange and difficult it is to classify computer programs and technical information as something other than speech."

David Touretzky, a computer-science professor at Carnegie Mellon University, has compiled a Web-based gallery that displays the critics' contributions to the cause. Samuel Hocevar, an engineering student in France, made a digital movie in which the DeCSS code scrolls off into space much like the introduction to "Star Wars." Two other programmers refined a way to transform the DeCSS code directly into music by assigning each letter, number and symbol its own musical note, with results that sound a bit like Philip Glass on acid.

DeCSS has inspired such passion because it has become a flashpoint in the war over the control of digital media. On one side stands Hollywood, which assumed that locking up DVD movies with sophisticated encryption codes would thwart piracy. On the other side are individual users, some doubtless eager to obtain pirated movies but others including civil libertarians who believe the movie industry's efforts could stifle encryption research.

By stripping away the encrypted copy protection on a disk, DeCSS makes it possible for users to treat DVD movies like any other digital file. When compressed by a second program called DivX, the movies can even be distributed over the Internet—a potential nightmare for Hollywood. That scenario is one reason the movie industry lobbied heavily for copyright-law changes that made it illegal to defeat copy protection with decryption "devices," a provision that played a large role in Judge Kaplan's ruling.

But libertarian-minded hackers argue that the

Please Turn to Page B6, Column 3

The Many Faces of DeCSS

The decision to ban a Web site from posting DeCSS, a software code that can break the copy protection on DVD movies, has spawned a variety of creative works that include the code or instruct an audience how to recreate it. The creations themselves can't be interpreted by a computer as code. Below, some of the interpretations

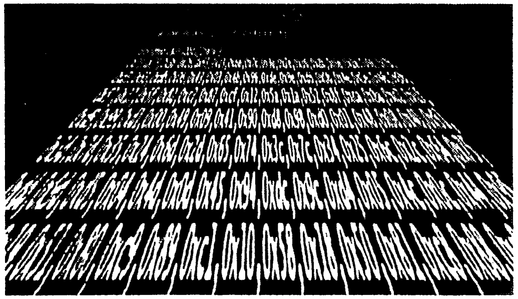
A Haiku

In this excerpt from a 456-stanza epic, the poet refers to Hollywood's DVD Copy Control Association

> "Now help me, Muse, for
> I wish to tell a piece of
> controversial math.
>
> For which the lawyers
> of DVD CCA
> don't forbear to sue.
>
> That they alone should
> know or have the right to teach
> these skills and these rules
>
> (Do they understand
> the content, or is it just
> the effects they see?)"

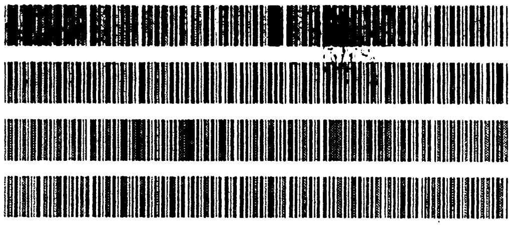
A Movie

In this digital movie by Samuel Hocevar and friends, the code scrolls into space like the introduction to Star Wars



A Bar Code

Using a simple cipher, the C source code has been translated into standard bar code by Scott C. Potter



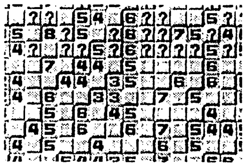
A Typo

This transcript of day six of the New York DVD trial is riddled with errors; the typos are an encryption of the code by Scott A. Crosby

> "THE COURT: Good morning.
> MR. KTLAS: Before we gegin with the teslimony today, would it be posgible to move in tPe exhibits and thN depositioM testimony, or woMld you rather do ghat before the break or ..."

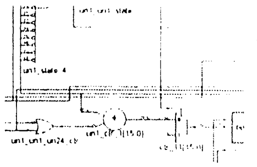
A Game

This portion of a playable, computer game board is also an encoding of the code



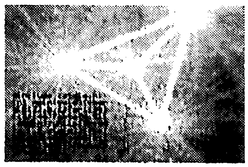
A Schematic

Part of a blueprint for building a device to circumvent DVD encryption



An Image

This image, created by Joshua Shagam, has the DeCSS code encrypted in the computer text version of the image



Liz Shura/The Wall Street Journal

Sources: Dr. David Touretzky; http://www.cs.cmu.edu/~dst/DeCSS/Gallery/index.html; http://www.cs.cmu.edu/~dst/DeCSS/Gallery/Stego/index.html

Chrysler M...
Ads Sayin...
It's Not De...

By Jeffrey B...
Staff Reporter of The Wall...

AUBURN HILLS, Mich....
Chrysler AG was born in 199...
vited consumers to "expect ...
Now the stumbling German-A...
Chrysler unit is aiming a diffe...
can consumers: We're not de...

Chrysler, beset by well-pul...
ink and management turmoil,...
ing out a marketing blitz this...
gallows humor to convince Am...
words of one proposed ad, the...
and kicking."

Driving the strategy are t...
were hired away from rival ...
February: James Schroer, ex...
dent of global sales and mark...
Murphy, senior vice presiden...
marketing. While some call t...
the Chrysler executives belie...
necessary to return the unit to...
question is not if we do it, but...
how we do it," Mr. Murphy ...
debate: How aggressively do...

Some of the early drafts o...
which sit on the windowsill o...
fice, are aggressive indeed. ...
headlines, side by side: "Ru...
Under "Fact" are photos of t...
cles, including its hot-selling...
"Rumor" is another famous ...

Among the other proposed...
out of a deep hole," which sh...
Liberty sport-utility vehicle a...
pickup truck; "It's true. We ...
list," which quotes media prai...
company's cars and trucks; "...
with pictures of the PT Crui...
Viper muscle car; and "Here ...
row," which shows both curre...
and concepts of future ones.

Chrysler executives still...
which of the ads, developed by...
of **Omnicom Group** Inc., to...
the end, the company probably...
keting messages to different a...
ers, dealers and employees. ...
phase in the edgier ads over...

The ad with the hearse, ...
wouldn't be appropriate to ru...
after Chrysler implemented ...
part of a corporate cost-cuttin...
rid of 26,000 jobs, or 20% of Chr...
But, he adds, "eight months...
we've hit the turnaround plan...
running again and people fe...

Please Turn to Page B1...

John J. Leahy, executive vice president for customer affairs at Airbus, its top salesman, predicted this year "will be down a bit," noting that airplane orders "tend to track" regional economic growth.

Airbus last year garnered orders for 520 planes and expects orders for between 350 and 400 planes this year. Boeing won 611 orders last year and expects fewer this year, but won't make a specific sales forecast. Both are currently expanding production to fill orders placed over recent years. Airbus's order backlog stood at 1,660 planes on March 31 and Boeing's stood at 1,583 planes.

Today's orders go into production months or years from now and deliveries stretch over many years, so a drop-off could take some time to be felt by the

... options to increase their orders. If demand is strong, they exercise their options; if things turn bad, they let the options expire. And if things get really bad, as they did a decade ago, buyers start telling the plane makers they want to postpone deliveries. That can throw production planning out the window.

Boeing says that all of this year's production capacity consists of firm orders and that 80% of next year's deliveries have already been set in stone. Airlines, depending on how many airplanes they order, can negotiate contracts that allow them to delay the decision to finalize orders to within 18 months of the delivery date. Typically, that decision must be made about two years before the airplane is delivered.

Mr. Belyameni said U.S. airlines are con-

While airlines' performance in the U.S. and parts of East Asia is weakening, global air traffic continues to rise. And although a sharp drop in the broader economy could still damp the cautiously optimistic outlook, several factors give the plane makers a cushion.

For one, there are lots of old, noisy and uneconomical planes still flying that need to be replaced. James Goodwin, chief executive of United Airlines' parent, UAL Corp., said recently that his orders from Boeing and Airbus are fixed through 2003 and "we don't see any need to alter" plans, although a sharp change in the situation would prompt a reassessment. Meanwhile, United is retiring old models such as Boeing 727s and DC-10s faster than it had planned to last year.

into recession a decade ago
ing powerhouses, which pla
for all types of planes and tl
out to airlines, now account
30% of industry orders. Mar
tomers are second- and thir
that can't afford to buy pl
and that often lack the fina
to weather a downturn.

"We're coasting in a fra
not accelerating or "falling
said John L. Plueger, execut
dent and chief operating offi
tional Lease Finance Corp.
leasing arm of insurer Amer
tional Group. "We're still pla
although sometimes it's a bi

—J. L

contributed t

Banned Program Lives in Poetry and Song

Continued From Page B1

DeCSS case isn't so much about the threat of piracy as about Hollywood's desire to dictate how its movies are distributed and viewed. Decoded DVD movies, for instance, no longer bear the "regional coding" designed to protect Hollywood's international film-release schedules by preventing non-U.S. viewers from playing U.S.-released DVDs. Some legal scholars also believe that the copy-protection provisions of copyright law are overly broad and possibly unconstitutional.

The fight over DeCSS goes to the heart of an emerging conundrum in free-speech and intellectual-property law. Software code, after all, is just a representation of mathematical ideas that can range from simple addition and subtraction to complex functions that paint graphics on computer screens, simulate weather patterns and decipher encryption systems, such as Hollywood's Content Scramble System (from which DeCSS took its name).

Like other ideas, these mathematical notions can be translated into and out of different specialized languages, from "high level" computer languages, such as C, Java and Perl, to the binary ones and zeroes in programs that computers can execute directly. Such ideas can even be described in plain English.

The DeCSS poem not only outlines the algorithm but includes six lengthy arrays of numbers used to help simplify the decryption calculation, all without breaking the syllabic 5-7-5 haiku form. At one point, the poem cites a six-digit "player key," a kind of master key to the CSS algorithm that Hollywood considers proprietary:

"So this number is,
once again, the player key:
(trade secret haiku?)

"Eighty-one; and then
one hundred three – two times; then
two hundred (less three)

two hundred twenty
four; and last (of course not least)
the humble zero."

Few average readers of the haiku would find it much help in breaking DVD encryption, but turning the poem back into software code wouldn't be difficult for most programmers. A sufficiently devoted hacker, in fact, might even automate the process by writing a program specifically designed to turn the DeCSS haiku back into a functioning program, further blurring the distinction between speech and code.

Judge Kaplan explicitly rejected the free-speech arguments made for DeCSS, noting that because the code is "functional" it can be regulated more strictly than other forms of expression. His ruling is now on appeal, and civil libertarians hope the constitutional argument will be more effective before the appellate court.

If Judge Kaplan's ruling is upheld, the consequences for the DeCSS code won't necessarily be clear. Legal scholars such as Yochai Benkler of New York University suggest that higher courts may uphold a ban on the executable form of the program, which can actually decrypt DVDs, but not its underlying software code or artistic works that incorporate it.

Hollywood, itself a big beneficiary of First Amendment freedoms, seems to take a similar position. "The fact is that many of the things Prof. Touretzky is talking about don't present the same kind of harm to content owners that the [executable DeCSS] utility does," says Charles Sims, who represented the movie studios before Judge Kaplan. The DeCSS haiku, he adds, "is not the sort of thing that content companies would spend money pursuing."

That seemingly reasonable compromise might not be the last word. Were a Web site to host any of the nonexecutable DeCSS works along with a simple program that converted them back into a functioning decryption program, the effect would be virtually identical to hosting the DeCSS program.

"If one is to be consistent, you have to ban the haiku," says Jonathan Zittrain, a Harvard University law professor sympathetic to the First Amendment argument for DeCSS. By Judge Kaplan's analysis, he says, "the underlying illegality is still illegal" no matter what form the code is in.

Students Protest Universities' AIDS-Research Profit

Continued From Page B1

moved on it," she says.

None of that deters Ms. Swarr, who recently returned from a year in South Africa, where she worked with AIDS activists from the global Treatment Action Campaign.

There are close ties between student activists and outside activists, and their work tends to reinforce each other. Activists at Yale—where the student protests against AIDS-drug prices began—were keyed up by a Feb. 14 letter sent to the university's Office for Cooperative Research by Doctors Without Borders, a humanitarian, not-for-profit group. The letter, drafted by activist Toby Kasper and signed by Eric Goemaere of Doctors Without Borders, asked Yale to use its patent on Zerit to pressure Bristol-Myers Squibb to lower the price of the medicine and release its patent rights in poor countries.

Amy Kapczynski, a first-year law stu-

cense to distribute Zerit. Already working on AIDS issues, she figured she had found the perfect cause. At the same time, Yale graduate students were holding meetings to discuss the ethics of academic and industry licensing agreements. They, too, were looking for an example to rally around and found it in a Yale Daily News article about Zerit and the license with Bristol-Myers.

By March 9, about 600 Yale students, faculty and researchers signed a petition demanding that Yale push Bristol-Myers to make the drug affordable. On March 15, Bristol-Myers Squibb became the first drug company to announce it would relinquish patent rights for an AIDS drug in South Africa. A spokesman says the company was in talks with Yale before the protests began and the students played no role.

Nevertheless, students and global AIDS activists are taking credit. Now they ...

200 students showed up for a teach-in on the New Haven campus.

In Minnesota, Ms. Swarr, spurred by the Yale protest, decided to take on her own administration. After seeing a March 12 article in the university's daily newspaper that mentioned the Ziagen patent, she contacted Zachie Achmat, a South African activist she knew. He put her in contact with the activists at Yale, who were then in the middle of their petition drive.

Within a week, Ms. Swarr says, the students at Minnesota were in contact "with almost every major international nongovernmental organization working on issues of affordable HIV/AIDS treatment." Oxfam, an activist group based in the United Kingdom, weighed in with a March 28 letter to the university's president, Mark Yudof, asking that the school hand over its Ziagen patent

Video-Game Fa
Get Cocky as I
Of Xbox Appı

Continued From Pag

interested in breaking news
the company line. John O'
crosoft's director of games m
knowledges that it has to resp
editorial control while trying
age them from publishing in
mors. "We strike a delicate l
says.

Game companies usually of
press releases, pictures and s
their PR firms. Microsoft has
sons for embracing fan sites
rivals, the company has never
player before. Although it has
to spending $500 million world-
moting the Xbox, it would
sense to tap that fat budget un
shipment this fall.

Right now, Microsoft offers
own Xbox.com site to fan sites
a set of guidelines. Requireme
making weekly updates of infor
refraining from "overtly negativ
tary about the Xbox." While
works with more than 30 Xbox
there may be dozens more out
ing listed on Xbox.com helped
the number of visitors to FunX
Calgary-based site operated by
rant, 16. "More visitors mean
revenue," he says.

There are other perks to coop
Sol Najimi, the 20-year-old edit
box.com has learned. A couple
fore Microsoft Chairman Bill Gat
to unveil the Xbox machine at a
electronics convention in Las Veg
6, MSXbox.com and other fan
played leaked photos of the ma
were scheduled to be published in
ing issue of Electronic Gaming
magazine. Within half an hour
posted the pictures, Microsoft as
remove them, Mr. Najimi recalls
says the sites that published t
were infringing on the magazi
right. Mr. Najimi says he obliged
cause Microsoft asked.

Microsoft had invited Mr. Najir
tends college in New York, and a c
come to the Las Vegas conventio
an online chat session on MSXbox
ing Mr. Gates's speech. The junk
included airfare and lodging at
Grand, cost Microsoft about $1,00
jimi estimates. The exclusive cha
box.com drew 500 people, or 10 tim

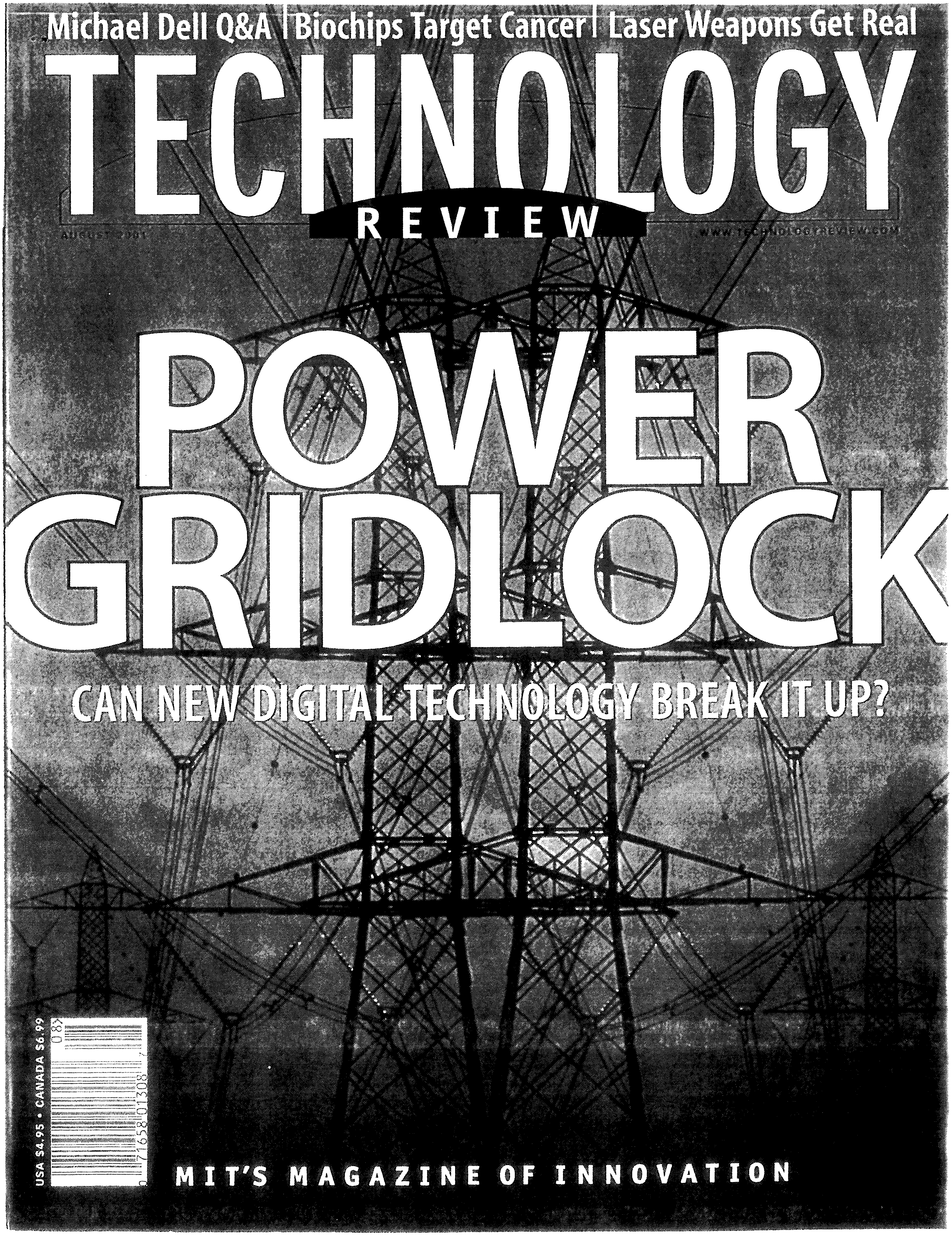
Ex. B to Ex. 3

Michael Dell Q&A | Biochips Target Cancer | Laser Weapons Get Real

TECHNOLOGY REVIEW

AUGUST 2001

WWW.TECHNOLOGYREVIEW.COM



POWER GRIDLOCK

CAN NEW DIGITAL TECHNOLOGY BREAK IT UP?

USA $4.95 • CANADA $6.99

MIT'S MAGAZINE OF INNOVATION

SIMSON GARFINKEL *The Net Effect*

The DVD Rebellion

BUY A COPY OF *THE MATRIX* on DVD and take it home. Play it on a Mac or on a Windows PC and you're in for a pretty good time. But play it on a PC running the Linux operating system, and the movie industry says that you're breaking the law.

Your transgression is that of "circumvention," a criminal act created by the 1998 Digital Millennium Copyright Act. You see, the video on DVDs is scrambled. Windows and Macintosh DVD players licensed by the DVD Copy Control Association contain the algorithms to unscramble the signal. The Linux DVD player contains these secrets as well. But since the Linux-based program isn't licensed, using the software constitutes an illegal circumvention of copyright management.

DUNG HOANG

Things have gotten nasty as this new crime gets its tryout in the legal system. Last year, eight major film studios, all members of the Motion Picture Association of America, sued the magazine *2600* for posting on its Web site a program that unscrambles DVDs. Not only did the organization win its case, but U.S. District Court judge Lewis A. Kaplan even barred *2600* from posting links to other sites that contained the program. That case is now on appeal.

For the movie industry, the DVD case is about piracy and revenue protection. For the programmers among us, the attempts to suppress this software are an attack on fundamental freedoms of speech and inquiry. It is a battle the movie industry is sure to lose. The only question is, "when?"

At the core of the controversy is technical data about the copy protection techniques used to make DVDs. The information on each DVD is protected by an encryption scheme called the Content Scramble System, or CSS. This technology prevents computer users from duplicating a movie, compressing it down to fit on a CD-ROM, and then giving copies to their friends. Playing the DVD entails decrypting the data—an act that used to require a licensed DVD player with the appropriate descrambling algorithms, stored either in a program or in a set-top box.

Then in 1999, an anonymous European programmer cracked the code and distributed a program—called DeCSS—over the Internet. Ever since, the movie industry has been filing lawsuits and sending threatening letters to individuals and businesses that distribute this and related DVD decryption programs.

How did we get here? In the 1980s, compact discs revolutionized high-fidelity sound. But CDs were not well suited for movies: their roughly 600 megabytes could store barely 10 minutes of video. (Advanced compression systems can put an entire movie on a CD, but the quality suffers.)

Enter DVDs, which can store more than two hours of compressed video on a disc the same size as a CD. If you want to make your own DVDs, you can buy a recording drive for less than $500. Rewritable discs that hold 4.7 gigabytes cost about $30.

It's easy to see why the movie studios are worried. The price of recordable DVD discs is sure to fall. Three years ago, writable CD-ROMs cost $2; today, they're 40 cents or less. Expect writable DVDs for $5 by mid-2002. Equipped with programs like DeCSS, consumers will be able to make high-fidelity copies of DVDs on the cheap.

Movie studios have long been terrified of home recording technology. In 1983, Sony and Universal City Studios

faced off over the legality of home videocassette recorders. Universal said VCRs should be outlawed because they could be used to make illegal copies of copyrighted materials. But in 1984, the U.S. Supreme Court ruled that "the sale of [VCRs] to the general public does not constitute contributory infringement of respondents' copyrights." The Court reasoned that recording a television show at one time for viewing at another fell under the "fair use" provision of copyright law.

For the movie industry, the issue is piracy. But for programmers, the attempt to suppress DVD unscrambling software is an attack on fundamental freedoms of speech and inquiry.

The movie industry has never been happy with this decision, and in 1998 it prevailed upon federal lawmakers to do something about it. Unable to overturn a Supreme Court ruling, Congress did the next best thing: it passed the Digital Millennium Copyright Act, which created the crime of "circumvention."

Copyright is supposed to balance the rights of publishers and the rights of the public, explains Cindy Cohn, legal director of the Electronic Frontier Foundation, a civil-liberties organization defending *2600* magazine. The new law, says Cohn, makes an end run around fair use by making it illegal for any person to use or distribute technologies that can circumvent a copyright protection system. Because of the public's right to fair use, Cohn says, "every time the content holders have tried to reach out and get more control, as they did with VCRs, the Supreme Court has slapped them down."

It isn't just movies that could fall under this new form of protection. Any company that wants to prohibit fair use can simply wrap its products—movies or books or magazine articles—in a thin layer of cryptography. The content purveyor could then apply restrictions that made it possible to view the material only by using the publisher's proprietary software. If the software doesn't allow fair-use rights, then the 1998 legislation makes it illegal for people to circumvent that software to get their rights back.

The original 1999 program that broke the DVD encryption algorithm was created not for piracy but to let people who bought DVDs play them on computers running Linux. But science marches on. In March 2001, two programmers at MIT reduced the original 215-line decryption algorithm to just six lines. It has become so small that people are putting it at the bottom of e-mail messages as a "signature." You can even purchase a T-shirt displaying the forbidden code. *Technology Review* would probably not consider printing a 215-line program; the six lines appear below. "The shorter the program gets, the sillier the studios look for trying to suppress it," says Carnegie Mellon computer scientist Dave Touretzky, who posts a gallery of DVD decoders on his Web site.

Another front in the DVD wars has opened up at Princeton University.

THE DVD UNSCRAMBLER

Six lines of code that have rattled the movie industry

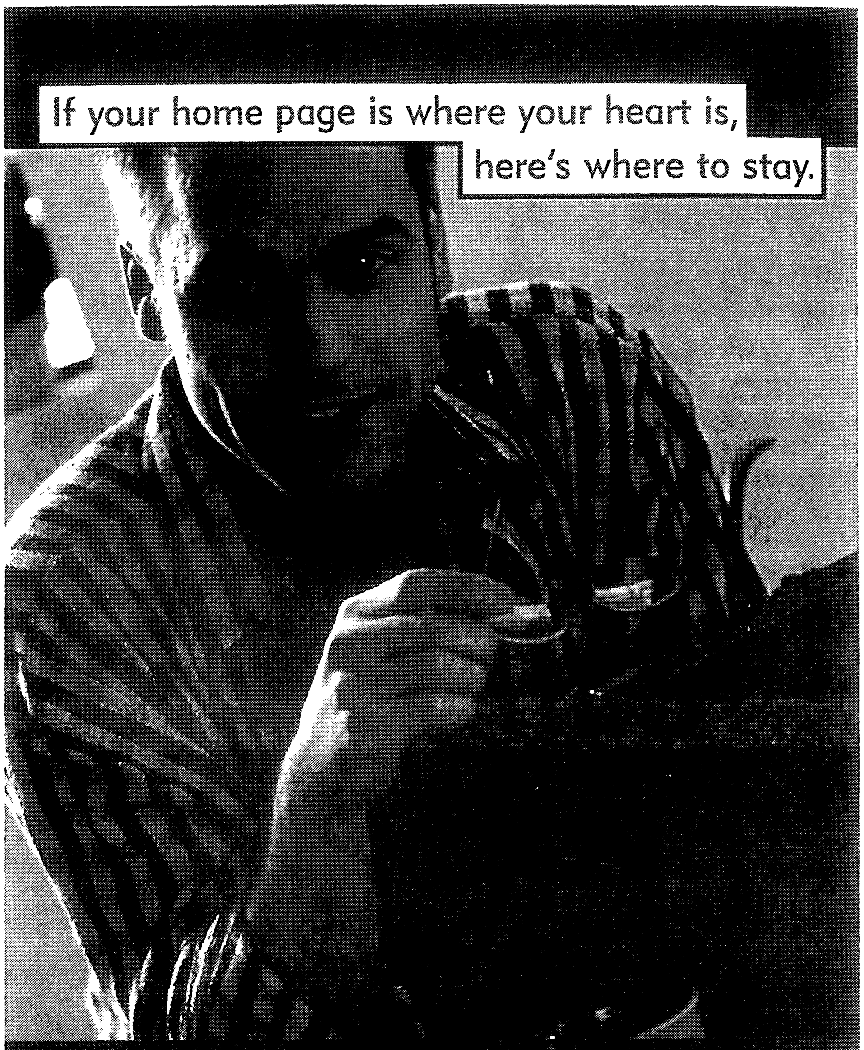
```
s''$/=\2048;while(<>){G=29;R=142;if((@a=unqT="C*",_)[20]&48){D=89;_=unqb24,qT,@
b=map{ord qB8,unqb8,qT,_^$a[--D]}@INC;s/...$/1$&/;Q=unqV,qb25,_;H=73;O=$b[4]<<9
|256|$b[3];Q=Q>>8^(P=(E=255)&(Q>>12^Q>>4^Q/8^Q))<<17,O=O>>8^(E&(F=(S=O>>14&7^O)
^S*8^S<<6))<<9,_=(map{U=_%16orE^=R^=110&(S=(unqT,"\xb\ntd\xbz\x14d")[_/16%8]);E
^=(72,@z=(64,72,G^=12*(U-2?0:S&17)),H^=_%64?12:0,@z)[_%8]}(16..271))[_]^((D>>=8
)+=P+(~F&E))for@a[128..$#a]}print+qT,@a}';s/[D-HO-U_]/\$$&/g;s/q/pack+/g;eval
```

In April, the Recording Industry Association of America sent computer science professor Edward Felten a chilling letter stating that Felten's publication of a paper on the Secure Digital Music Initiative's watermarking algorithm might constitute a criminal act. Felten pulled the paper from its scheduled release at a conference. Since then, however, it has been all over the Internet.

All this recalls what happened back in the 1990s in response to the Clinton administration's absurd restrictions on cryptography. Strong encryption was classified as munitions; exporting crypto was punishable by up to 10 years in prison and up to a $1 million fine. So programmers reduced a powerful encryption algorithm known as RSA to three lines of code and plastered it all over the Web; at least three people even had the lines tattooed onto their bodies. In 1997, the U.S. Department of Commerce decreed that exporting this potent snippet of text required a license. Not that it mattered. Two years later, the administration caved.

Mark Litvack, an attorney representing the MPAA, insists "it has not been our intention to stop debate on the merits and values" of DVD encryption. Instead, he says, his organization is merely trying to wipe out Web sites that are distributing illegal "circumvention devices." Thus, in February, the association wrote a letter to Carnegie Mellon demanding that the university remove Touretzky's Web pages from university servers. The university did not comply.

The movie industry lost its battle over DVDs when it decided it would be neat to let people play DVDs not just on TV sets but on computers. There's no way to keep secret something that's distributed to millions of PC users. Information is power, and computers are machines designed to process and distribute information. Moviemakers are about to learn what the Clinton administration learned with crypto: no matter how you legislate, information wants to be free.



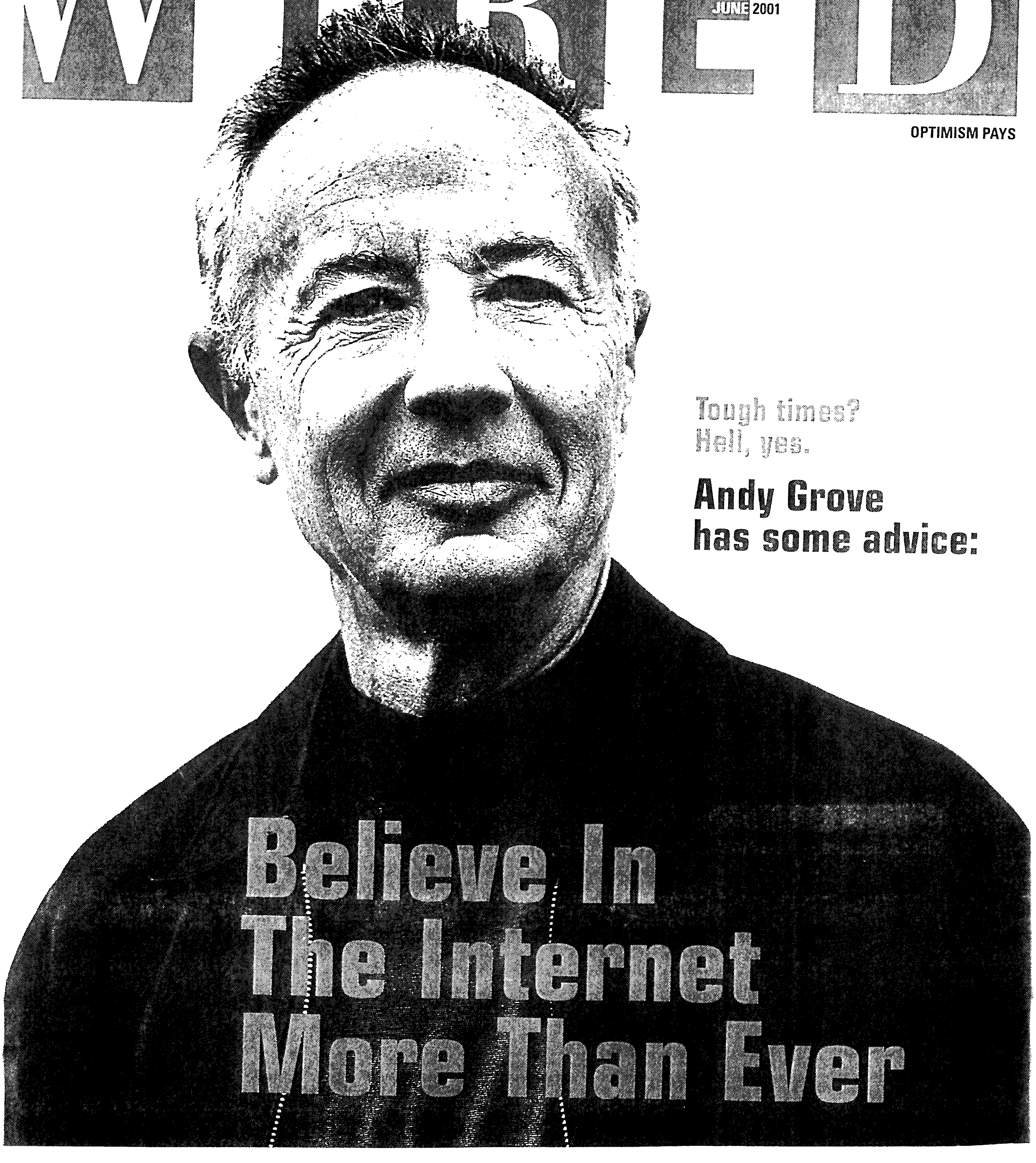
Ex. C to Ex. 3

Gurri's copy

WIRED

JUNE 2001

OPTIMISM PAYS



Tough times?
Hell, yes.

**Andy Grove
has some advice:**

Believe In The Internet More Than Ever

Gorilla Positioning System

REMOTE SENSING Tracking mountain gorillas in the volcanic jungles of central Africa isn't easy. If you avoid malaria, there's still the snipers. A newly forged partnership hopes to lessen the risks by allowing scientists to study the endangered beasts from desktops rather than treetops.

Rwanda is home to about 350 of the estimated 600 remaining gorillas. A typical trek into the country's Parc National des Volcans to view the animals requires five armed soldiers, two machete-toting trackers, and two local guides. So, software engineers and primatologists at Georgia Tech and the National University of Rwanda are teaming up with the Dian Fossey Gorilla Fund International to deploy a geographic information system developed by the Environmental Systems Research Institute of Redlands, California. From Rwanda, GIS data is sent via email, CD-ROM, and other means to Georgia Tech, where engineers have built a three-paneled screen to plot the gorillas' movements. A second layer of information is added by GPS and remote-sensing software, which generates hyperspectral data, or hi-res images, in varying light wavelengths that are rendered into 3-D maps. Eventually, wireless technology will let trackers in the field instantly send GPS coordinates over a local cellular network to a Web-linked database.

"Data from the satellites can ascertain the types of vegetation in the habitat and estimate how much gorilla food exists in the area," says Nickolas Faust, a principal research scientist and associate director at Georgia Tech's Center for Geographic Information Systems.

"Up to 10 researchers in Georgia and 5 in Rwanda are working on the project," says Faust, "and we intend to expand the effort significantly this year." The project has nearly $1 million in support from donors, including the Turner Foundation, the National Geographic Society, and Oracle. Safari Bonfils, dean of the Faculty of Science and Technology at NUR and director of the Rwandan side of the project, says he wants to expand the project beyond the gorilla. "We'd like to monitor agricultural development, land use, erosion, and mining exploration." - *Michael Behar*

Spying on Silverback in Rwanda from Georgia Tech.

DVD Hacking for Dummies

DECRYPTION Today's assignment: Descramble a DVD in less than seven lines of Perl. The lesson? Something this simple is more like a recipe for chicken soup than a circumvention device, says Keith Winstein, who solved this brainteaser with Marc Horowitz back in December.

Their efforts, called qrpff, have been copied from .sig files and Web posts worldwide, and the duo – members of the MIT Student Information Processing Board – have been pegged as mischief makers ready to take on the Motion Picture Association of America. Winstein, however, would like to set the record straight: "We're not saying, 'Screw you, MPAA; all intellectual property should be free.' We wanted to add to the public debate about whether or not six lines of text can become a circumvention device, and we wanted to see how neatly we could do it."

To keep the discussion balanced, Winstein invited industry execs to his MIT seminar, "Decrypting DVD." David Barr, lead engineer at C-Cube Microsystems, a member of the DVD Copy Control Association, gave an independent overview about US copyright law. Jack Valenti, president and CEO of the MPAA, replied with regrets. "Valenti sent me a nice RSVP saying he likes 'entering the lion's den' but that he couldn't make it," explained Winstein. – *Victor C. Clarke*

```
s''$/=\2048;while(<>){G=29;R=142;if((@a=unqT="C*",_)[20]&48){D=89;_=unqb24,qT,@
b=map{ord qB8,unqb8,qT,_^$a[--D]}@INC;s/...$/1$&/;Q=unqV,qb25,_;H=73;O=$b[4]<<9
|256|$b[3];Q=Q>>8^(P=(E=255)&(Q>>12^Q>>4^Q/8^Q))<<17,O=O>>8^(E&(F=(S=O>>14&7^O)
^S*8^S<<6))<<9,_=(map{U=_%16orE^=R^=110&(S=(unqT,"\xb\ntd\xbz\x14d")[_/16%8]);E
^=(72,@z=(64,72,G^=12*(U-2?0:S&17)),H^=_%64?12:0,@z)[_%8]}(16..271))[_]^((D>>=8
)+=P+(~F&E))for@a[128..$#a]}print+qT,@a}';s/[D-HO-U_]/\$$&/g;s/q/pack+/g;eval
```

Here's Perl master Mark-Jason Dominus' dissection of the contentious qrpff script:

- **The name itself – qrpff – is "deCSS" encoded with the well-known rot-13 function. The script decodes the content scramble system (CSS), thanks to an Achilles' heel – the linear feedback shift register (LFSR), which produces data that looks random but isn't. CSS uses two LFSRs, and their output, combined with the encrypted data on the DVD, produces the original video data.**
- **Each 2-Kbyte sector contains a key to initialize the LFSRs. This code extracts the sector key and decrypts it by combining it with a title key. A DVD player has a secret player code that lets it read the title key. The qrpff user must supply the title key on the command line, represented by @INC.**

This sets up the LFSRs with sector-key data.

- **This section gathers the outputs of the two LFSRs.**
- **The decryption process also involves replacing certain bytes with others, according to a table. This code computes the table.**
- **The result from the table is combined with the LFSR output to decode the original byte value of the video data.**
- **The table lookup and LFSR step are performed for each byte of data in the sector, and the result is output as the original video data. The main loop of qrpff repeats the decryption for each 2-Kbyte sector of video data.**

PHOTOGRAPH: JACQUELYN BEHAR

Ex. 4

RICHARD R. WIEBE (SBN 121156)
425 California Street, Suite 2025
San Francisco, CA 94104
Telephone: (415) 433-3200
Facsimile: (415) 433-6382

THOMAS E. MOORE III (SBN 115107)
TOMLINSON ZISKO MOROSOLI & MASER LLP
200 Page Mill Road, Second Floor
Palo Alto, CA 94306
Telephone: (650) 325-8666
Facsimile: (650) 324-1808

ALLONN E. LEVY (SBN 187251)
HS LAW GROUP
210 N. Fourth St., Suite 201
San Jose, CA 95112
Telephone: (408) 295-7034
Facsimile: (408) 295-5799

ROBIN D. GROSS (SBN 200701)
ELECTRONIC FRONTIER FOUNDATION
454 Shotwell Street
San Francisco CA 94110
Telephone: (415) 436-9333
Facsimile: (415) 436-9993

Attorneys for Defendant ANDREW BUNNER

SUPERIOR COURT OF THE STATE OF CALIFORNIA

COUNTY OF SANTA CLARA

DVD COPY CONTROL ASSOCIATION, INC., Plaintiff, v. ANDREW THOMAS MCLAUGHLIN; ANDREW BUNNER; et al., Defendants.	Case No. CV - 786804 **DECLARATION OF COMPUTER SCIENTIST GREGORY KESDEN** **IN SUPPPORT OF DEFENDANT ANDREW BUNNER'S MOTION FOR SUMMARY JUDGMENT**

I, Gregory Kesden, declare:

1. I am a Lecturer in the Computer Science Department of Carnegie Mellon University in Pittsburgh, Pennsylvania. Among the courses I teach is the department's course in Operating System Design and Implementation. This course is one of the core courses of the Computer Science Department and is the department's most intensive course; it receives 18 units of credit while all other courses receive 12 units or fewer.
2. Issues of computer security and protection, including an introduction to cryptography, are an integral part of a modern operating systems course – and are becoming a more compelling issue each day. All of the major operating systems texts include coverage of this area.
3. As part of my course in Operating System Design and Implementation, I teach my students about information security and protection schemes and the potential vulnerabilities of such schemes. I also teach them about the ways in which reverse engineering is used to enable programs and data to operate compatibly with many different operating systems. In my teaching, I illustrate these concepts using information about the Content Scrambling System ("CSS") used to encrypt DVD movie disks.
4. Last fall I reorganized my Operating System Design and Implementation course to increase the lecture time of the course. The additional lecture time was used to expand the course's coverage of protection and security, networks, and the implementation of the operating system Linux, as well as other areas. As part of my overall revision of the course, I introduced material about CSS. Attached as Exhibit A are my lecture notes and slides I used when I taught CSS's algorithms and keys as part of my Operating System Design and Implementation course in the Fall 2000 Term. These materials are also available on the Internet at http://www-2.cs.cmu.edu/~dst/DeCSS/Kesden/index.html.
5. I selected CSS because it is a simple, understandable example of a stream cipher that exhibits some classic cryptographic techniques. Additionally, it is a useful example because it has some well-known and reasonably understandable vulnerabilities and exploits. CSS is a weak encryption system vulnerable to a number of different

cryptological attacks. By teaching how the CSS algorithms and keys operate, I am able to demonstrate how these attacks function. Students are always excited to learn about weaknesses in real-world systems – it makes them feel more expert than the experts. But, beyond that, it helps drive home a very important lesson for future systems developers – cryptography is hard and the process of developing a cryptosystem should be careful and the system thoroughly validated before it is implemented.

6. CSS, DeCSS, and other DVD descrambling programs also illustrate concepts of interoperability—the use of computer data and programs with many different operating systems. For example, because no authorized DVD player was available for the popular Linux operating system, a version of DeCSS as well as other DVD descrambling programs have been created for Linux. Without these programs, it was impossible to play authorized, original DVD movie disks on Linux computers.

7. I also gave a lecture about CSS and DeCSS at the University of California, San Diego, in the Spring of 2001.

8. CSS and its algorithms and keys are widely known in the computer science community, as are DeCSS and other DVD decryption programs. I was able to find on the Internet the information about CSS and DVD decryption I needed for my course. For example, Frank Stevenson's well-known paper analyzing CSS, a copy of which is attached as Exhibit B, is readily available on the Internet. DVD decryption information is also available in more tangible forms as well. Attached as Exhibit C are photographs of a DVD decryption program (in the Perl computer language) printed on self-adhesive stickers which were widely posted on the Carnegie-Mellon University campus.

I, GREGORY KESDEN, declare under penalty of perjury under the laws of the State of California that the foregoing is true and correct.

Dated: 11-26-Ø1

Gregory Kesden

Ex. A to Ex. 4

Lecture 33 (Wednesday, December 6, 2000)

Course: 15-412 Operating Systems: Design and Implementation
Lecturer: Gregory Kesden

Slides Used In Today's Lecture

CSS (ppt)

Content Scrambling System (CSS): Introduction

You may recently have heard about the Content Scrambling System (CSS) or CSS-compatible open-source software known as DeCSS. CSS, which includes both player-host mutual authentication and data encryption, is used to protect the content of DVDs from piracy and to enforce region-based viewing restrictions. It may also have other purposes and certainly has other side-effects. DeCSS, which is open source, allows Linux-based systems to access the content of DVDs by emulating a licensed player and performing the authentication and decryption.

The national attention is the result of a recent federal circuit court ruling, in which the court held, among other things, that the distribution of the DeCSS source code violates the Digital Millenium Copyright Act (DMCA). It is my understanding that this ruling holds that it is illegal to distribute CSS compatible source code, including by URL reference ("link") to the same.

Others, including myself, believe that there exists a very clear distinction between human ideas expressed unambiguously and concisely in source code and the machine that exists after this source code is compiled or interpreted, and made runnable within a machine's memory. I believe that ideas expressed in this source code are Constitutionally protected speech, and that only the executible machine may be considered a "circumvention technology". As we discussed on the first day of class, "A program is a specification, whereas a process is an instance of a program in execution."

Furthermore, I believe that reverse-engineering for the purposes of ensuring compatibility, as a matter of public policy, and as a matter of the DMCA, should be and is protected. I believe that DeCSS, the product of reverse-engineering, is nothing more than a tool which allows Linux boxes to run .VOB programs on equipment other than that licensed by the monopoly *DVD Copy Control Association.* But my opinion on this issue is uneducated and unreliable -- I am not an attorney, and at least one court seems to have taken a different view.

Although I may disagree with the law, I cannot stand in front of this classroom and violate it or suggest that you violate it. Instead, I encourage you to obey it -- and to act to change it if you disagree with it.

As we'll discuss shortly, CSS is based on two Linear Feedback Shift Registers (LFSRs). Although very efficient in hardware, LFSRs are somewhat inefficient to implement in software, and there are some interesting programming techniques that are used to speed up the process. Unfortunately, I do not believe it is lawful for me to show that code, or equivalent code, to you today -- for that reason, I will not. Unfortunately, the same concern leads me to suggest that you **refrain** from reviewing the DeCSS source code after this lecture. Unlike my suggestion to review an implementation of DES, obtaining source code for DeCSS may well violate federal law. So today, you will literally get, "As much of an education as the law allows -- and no more."

But don't take anything I've said as legal advice -- I am a teacher, not a lawyer. Ask an experienced and licensed attorney, if you have any concerns.

So, without further delay -- let's dig in and take a look at CSS as an example of a stream cipher and an authentication protocol.

System Overview

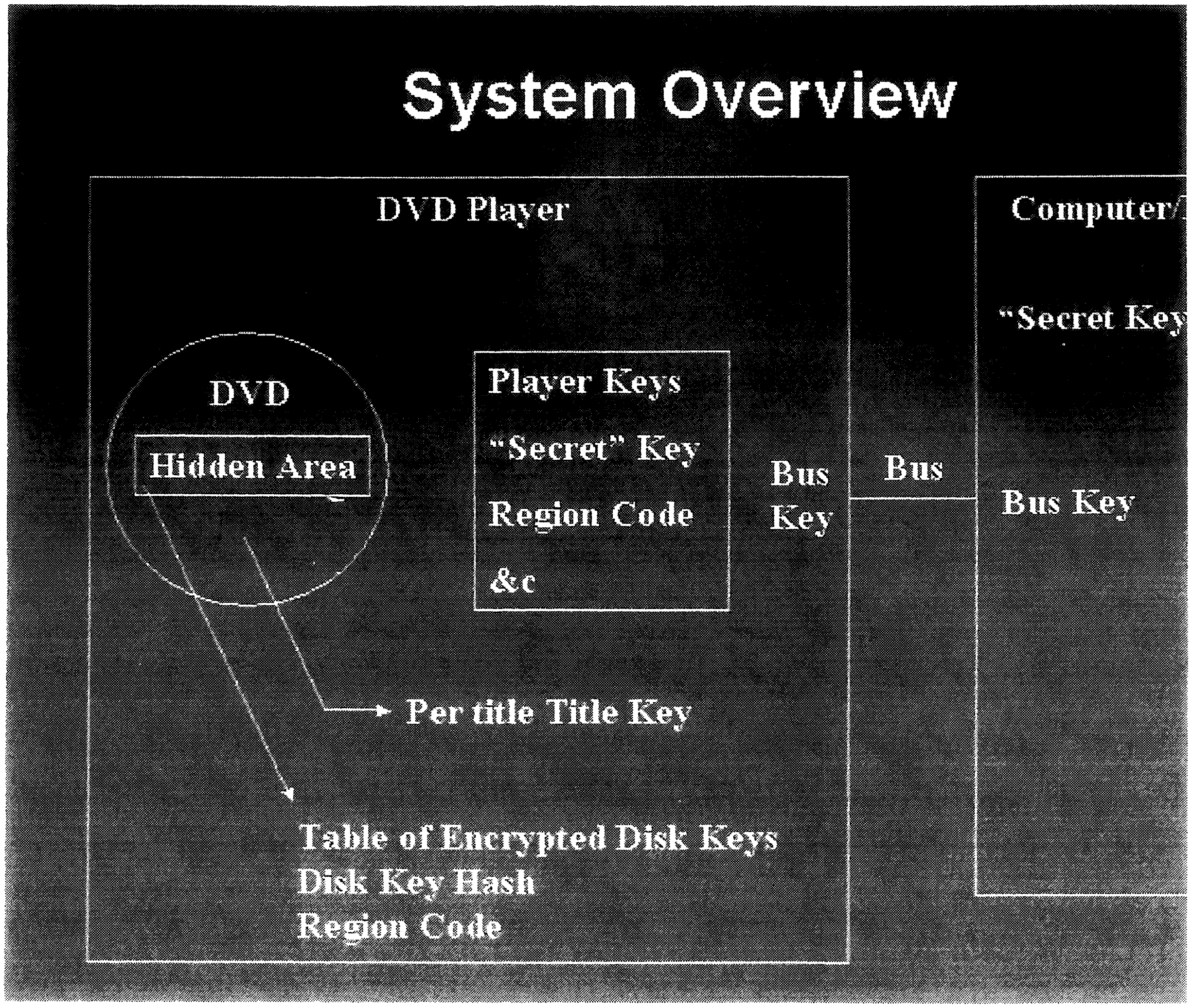
In our discussion of CSS, we are going to look at the system used to play DVDs in terms of three components: the DVD itself, the DVD player that reads the disk and delivers the content, and the host (computer, host board, &c).

The DVD disk itself contains the encrypted content, as well as a hidden area. It is my understanding that commerciallyt writeable DVDs already have this area marked, so that they cannot write to it. The contents of this hidden area cannot be delivered, except to an authenticated device. Presumedly, any device which can authenticate has been licensed by the *DVD Copy Control Association*, and as a consequence is trusted to receive the information. This hidden area contains the several pieces of information that we will soon discuss: a table of encrypted disk keys, and an encrypted disk key (disk key hash).

The player itself stores the player keys that are used to decrypt the disk key (more later), the region code that identifies the region in which the player should be used, and another secret that is used for authentication with the host.

The host seems to contain a secret that is used for authentication. It seems that this isn't a public key encryption scheme, but rather a private key scheme. We'll discuss authentication more shortly.

System Overview



DVD Player

Computer/

"Secret Key

DVD

Hidden Area

Player Keys

"Secret" Key

Region Code

&c

Bus Key

Bus

Bus Key

Per title Title Key

Table of Encrypted Disk Keys
Disk Key Hash
Region Code

Region Code

One other detail:

- Each DVD contains a region code that indicates the region of the world in which it is intended to be viewed.
- Each player knows the region in which it was to be sold.
- If the region code of the player doesn't match the region code on the DVD, the player won't deliver the data.
- This is to help the MPAA ensure that DVDs don't leak out into parts of the world ahead of the "first showing", &c.

Overview of Keys

Authentication Key

This "secret" is used as part of the mutual authentication process.

Session Key (Bus Key)

- This key is negotiated during authentication and is used to encrypt the title and disk keys before sending them over the unprotected bus. The encryption is necessary to prevent eavesdropping.

Player Key

- This key is Licensed by the *DVD Copy Control Association* to the manufacturer of a DVD player. It is stored within the player. It is used to establish the trustworthiness of the player. It is used to decrypt the disk key.

Disk Key

- This key is used to encrypt title key. It is decrypted using the player key.

Sector Key

- Each sector has a 128-byte plain-text header. Bytes 80 - 84 of each sector's header contain an additional key used to encode the data within the sector.

Title Key

- This key is XORed with a per-sector key to encrypt the data within a sector

Overview of the Process

Step 1: Mutual Authentication

- The host and the drive use a challenge-response system to establish their trustworthiness to each other. In the process, they negotiate a session key.

Step 2: Decoding disk

- The DVD player tries each of several player keys until it can decode the disk key. The disk key is a disk-wide secret.

Step 3: Send disk and title keys

- The title and bus keys are sent from the player to the host. The session key is used to encrypt the title and disk keys in transit to prevent a man-in-the-middle attack.

Step 4:

- The DVD player sends a sector to the host.

Step 5:

- The host decodes the title key using the disk key.

Step 6:

- The host decodes the sector using the title key, and a the sector key in the sector's header.

Disk and Player Keys

As we discussed, each player has a small number of licensed player keys. These keys can be used to decrpyt the disk key on a particular DVD. This disk key is used to decrypt title keys on the disk. Each work on the disk is encrypted with a title key. So in order to decrpyt the work, we must begin by decrypting the disk key.

This disk key is stored on the hidden sector of the DVD along with a a table containing the disk key encrypted will each of the 409 possible player keys. It also holds the disk key encrypted with the disk key.

The player decrypts the appropriate entry in the table and then verifies that it has correctly decoding the disk key, by decoding the encrypted disk key. The result, should be the disk key. That is to say that the decryption of the disk key, using the disk key, should prove to be the identity function. Players have more than one player key, so if the operation fails, they try again with an alternate.

Linear Feedback Shift Registers (LFSRs) and Encrpytion

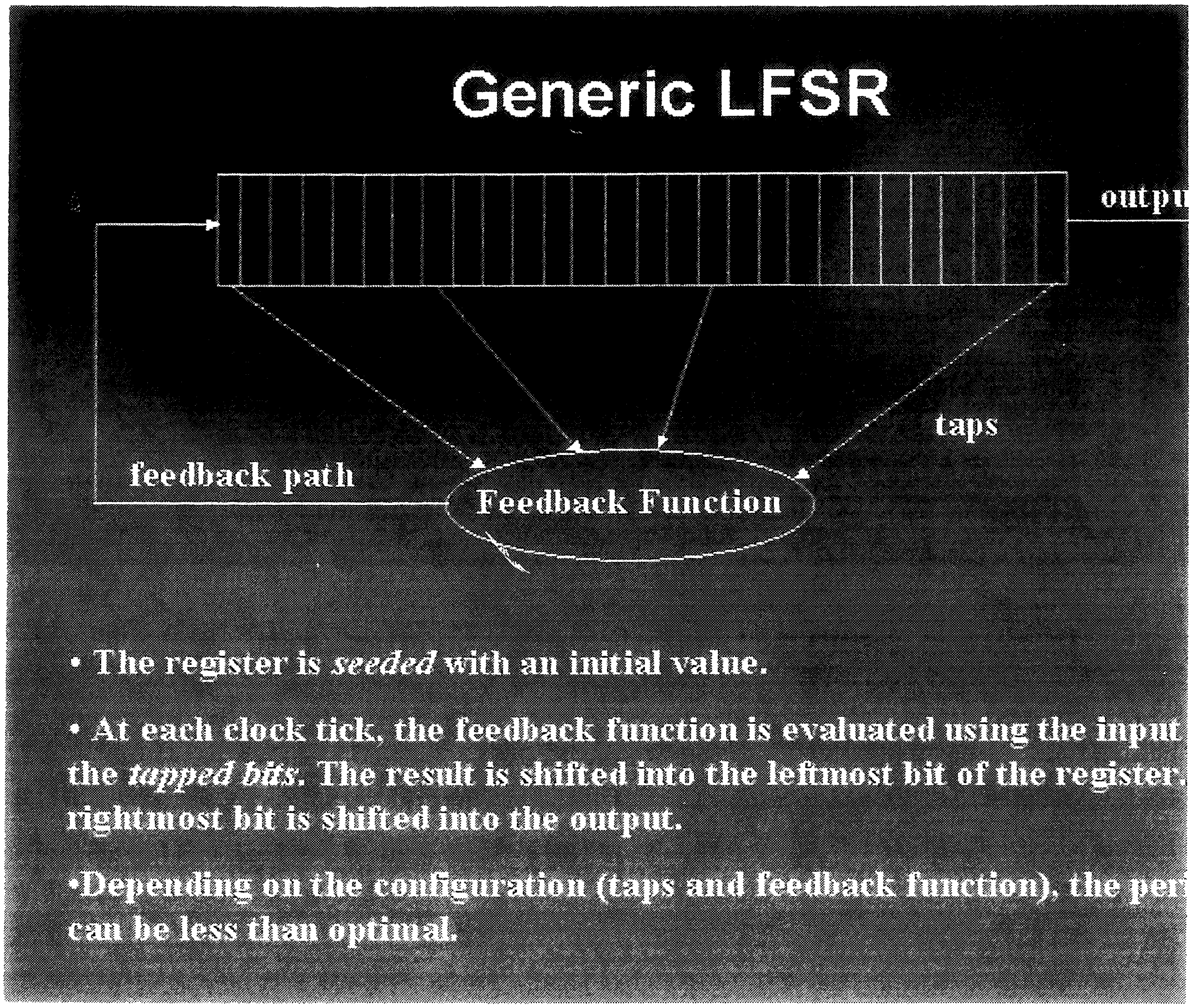
So, let's begin to talk about how the encryption works. We're going to begin with a little bit of background, then look at how data is encrypted, and then look at how keys, such as the title key above, are encrypted.

One technique used to encode a stream is to XOR it with a pseudo-random bit stream. If this random-looking bit stream can be regenerated by the receiver of the message, the receiver will be able to decode the message by repeating the XOR operation.

The LFSR is one popular technique for generating a pseudo-random bit stream. After the LFSR is seeded with a value, it can be clocked to generate a stream of bits. Unfortunately, LFSRs aren't truly random. They are periodic and will eventually repeat. In general, the larger the LFSR, the greater its period. The period also depends on the particular configuration of the LFSR. If the initial value of an LFSR is 0, it will produce only 0s, this is sometimes called *null cycling.*

LFSRs are often combined through addition, multiplexers, or logic gates, to generate less predictable bit streams.

An LFSR is *seeded* with an initial value. With each clock tick, certain *tapped* bits of the LFRS are evaluated by a *feedback function.* The output of this feedback function is then shifted into the register. The output of the register is the bit that is shifted out. This process is illustrated in the slide below:



Generic LFSR

- The register is *seeded* with an initial value.
- At each clock tick, the feedback function is evaluated using the input
the *tapped bits*. The result is shifted into the leftmost bit of the register,
rightmost bit is shifted into the output.
- Depending on the configuration (taps and feedback function), the peri
can be less than optimal.

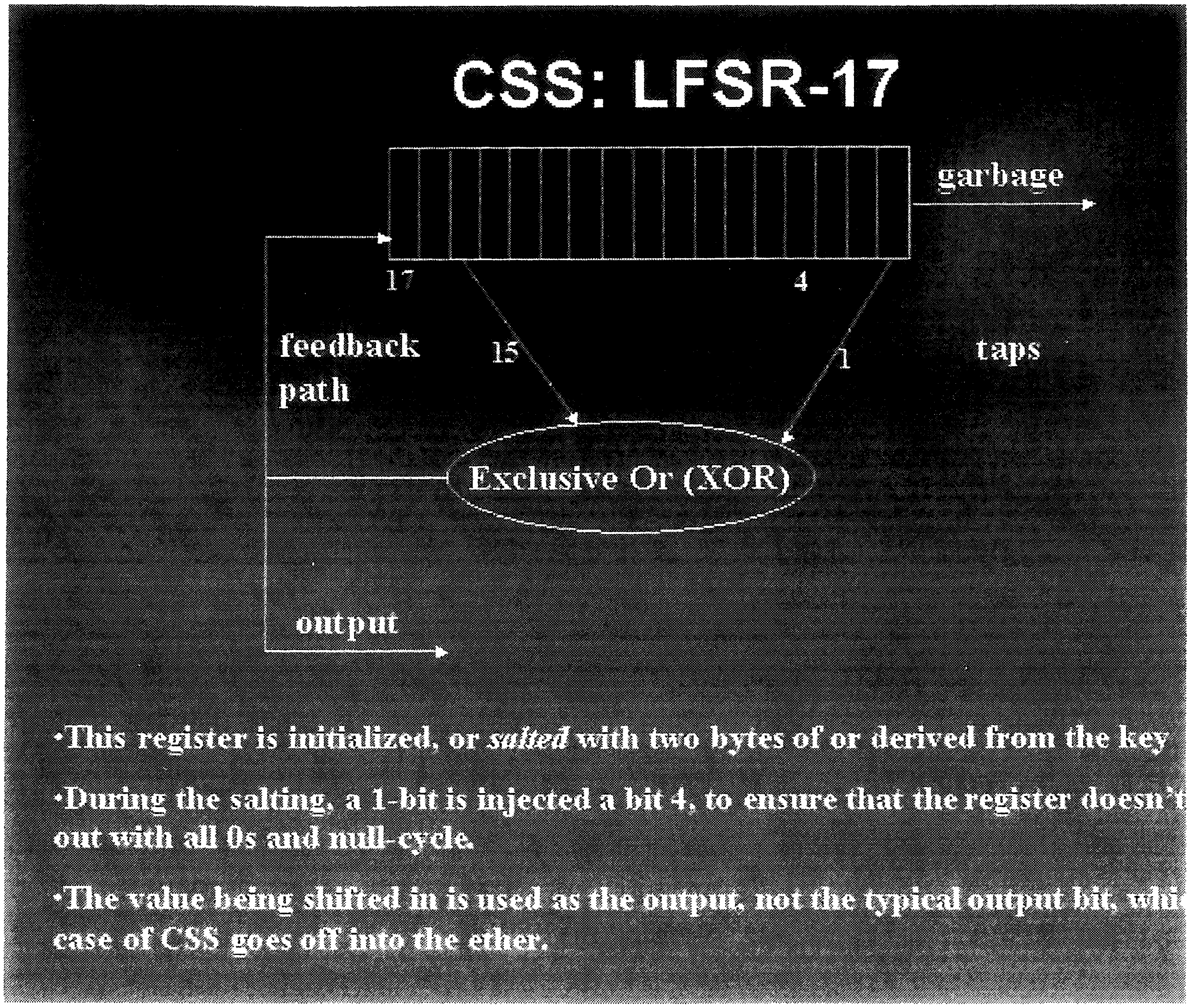
CSS's LFSRs

The CSS algorithm makes use of two LFSRs. The first is a 17-bit LFSR. Initially, it contains a two byte seed, with a 1 injected into the fourth bit, for a total of 17 bits. The is placed into the register to prevent null cycling. The second LFSR operates the same way, except it holds 25 bits.

Unlike typical LFSR-based stream ciphers, CSS throws away the bit that is shifted out of each LFSR. Instead, it considers the output of the feedback function to be both the input to the LFSR and the output.

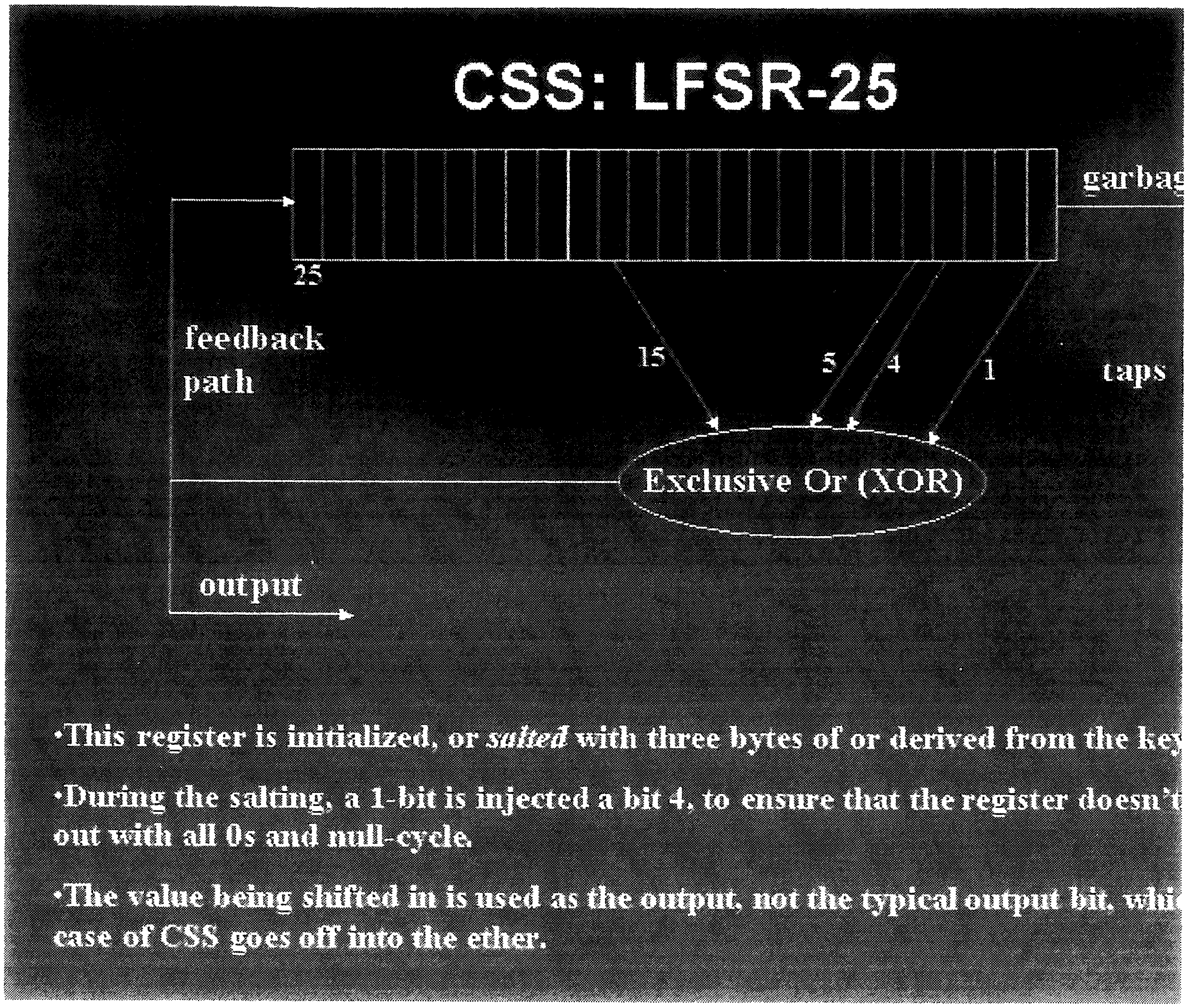
CSS uses a 40-bit, or 5 byte key. This is explains the size of the two registers: one is seeded with the first two bytes of the key, and the other the remaining three bytes of the key.

The two LFSRs are shown in the slides below:



CSS: LFSR-17

- This register is initialized, or *salted* with two bytes of or derived from the key
- During the salting, a 1-bit is injected a bit 4, to ensure that the register doesn't out with all 0s and null-cycle.
- The value being shifted in is used as the output, not the typical output bit, whi case of CSS goes off into the ether.



CSS: LFSR-25

- This register is initialized, or *salted* with three bytes of or derived from the key
- During the salting, a 1-bit is injected a bit 4, to ensure that the register doesn't out with all 0s and null-cycle.
- The value being shifted in is used as the output, not the typical output bit, whi case of CSS goes off into the ether.

LFSR Addition

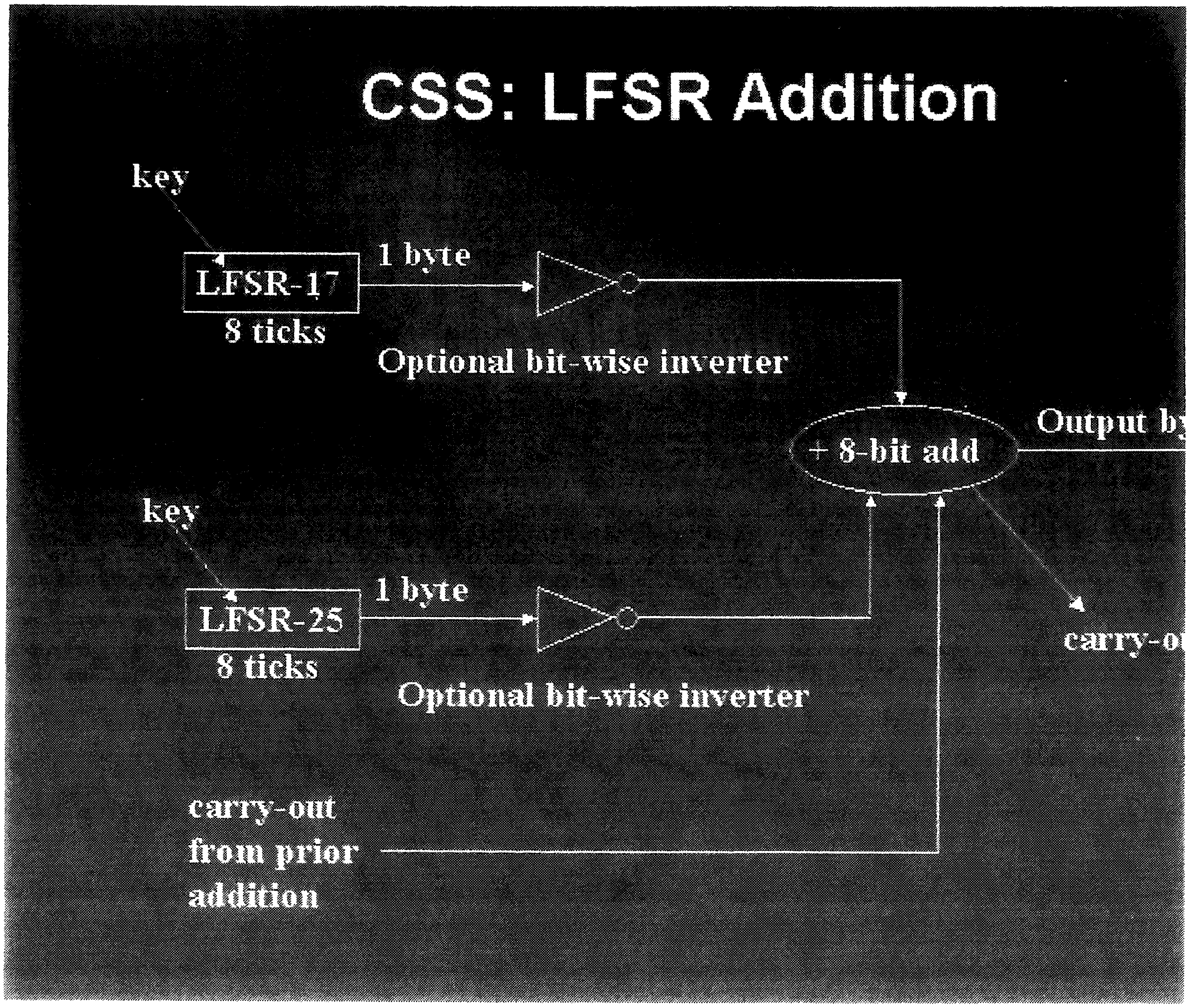
The output from the two LFSRs is combined using 8-bit addition. After each LFSR clocks out 8 bits of output, this output is added to form an output byte. The carry out from this addition is used as the carry in for the addition yielding the next output byte. It is worth noting that this is a pretty week way of using the LFSRs. Other approaches use more LFSRs, and do more complicated things with them, including clocking them at different rates, or combining them using multiplexers -- but not here.

CSS actually has four different modes. Depeding on the mode, the output of either or both LFSRs may be bit-wise inverted before the addition. The table below shows the inverter settings for each mode:

Invert Output of LFSR?		
Mode	LFSR-17	LFSR-25
Authentication	Yes	No
Session Key	No	No

Title Key	No	Yes
Data	Yes	No

The slide below shows the process, including the LFSRs, inverters, and addition. Please remember that each inverter is only enabled in those modes noted as "yes" in the table above:



CSS: LFSR Addition

key
LFSR-17
8 ticks
1 byte
Optional bit-wise inverter
+ 8-bit add
Output b
key
LFSR-25
8 ticks
1 byte
Optional bit-wise inverter
carry-o
carry-out from prior addition

Data Encryption/Decryption

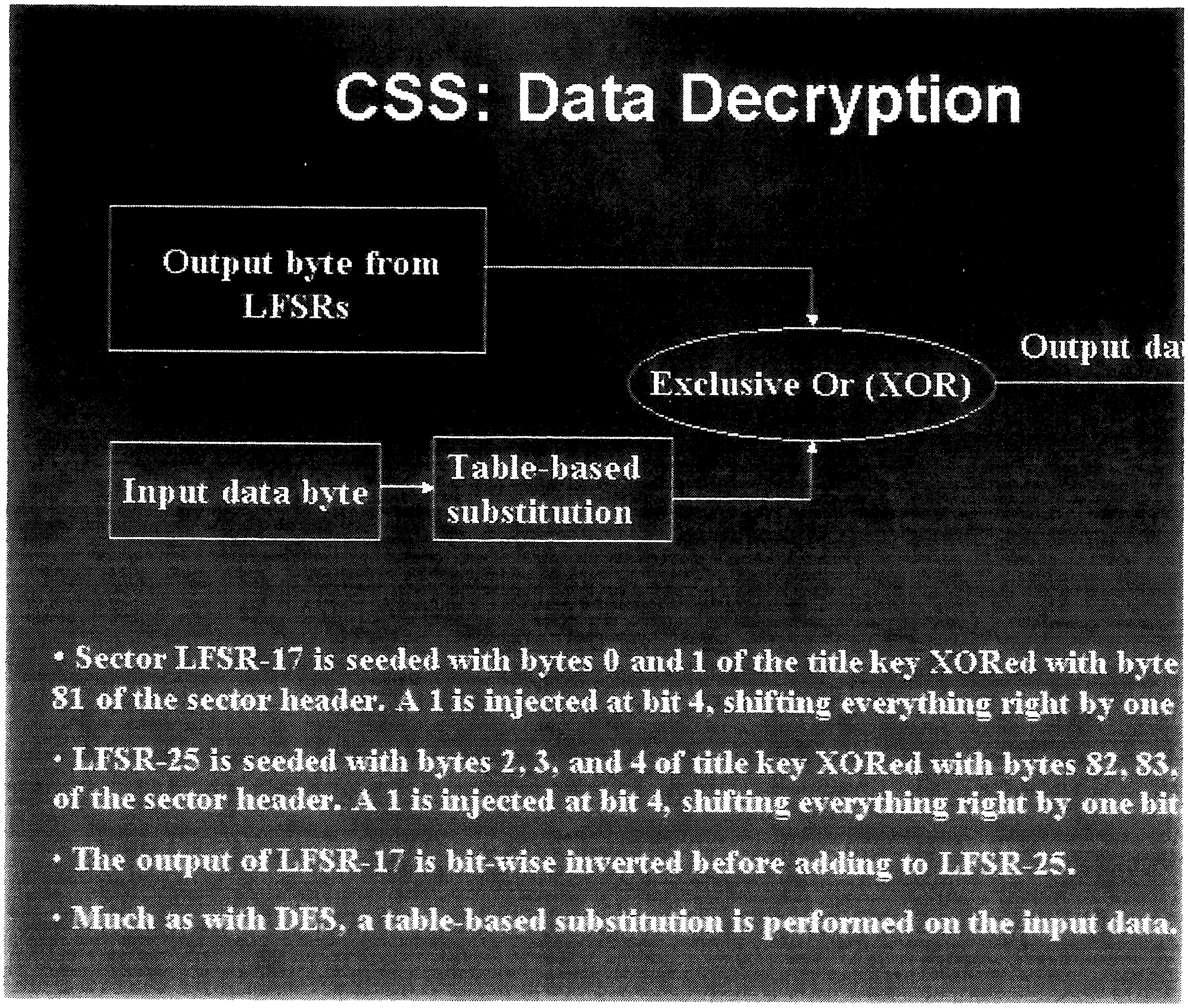
To encrypt or decrypt data, each LFSR is seeded with a portion of the title key. LFSR-17 is seeded with bytes 0 and 1 and LFSR-25 is seeded with bytes 2, 3, and 4. These bytes are seeded with a nonce that I called the *sector key* that is read from each sector.

The sector key is stored in bytes 80-84 of the sector. The first 128 bytes of each sector, the sector header, which includes the sector key, is plain text. The first two bytes (0 and 1) of the title key are XORed with the first two bytes of the sector (80 and 81), before seeding LFSR-17. Similarly, bytes 2-4 of the title key are XORed with bytes 82-84 of the sector, before seeding LFSR-25. Please also remember that a "1" is injected into each seed at bit 4.

to make the seeds 17 and 25 bits, respectively.

Once the LFSRs are seeded, their output can be added together as described above, to form the pseudo-random bit stream. This bit stream is XORed with the plaintext, to generate the ciphertext. Much as was the case with DES, bytes of the plaintext are run through a table-based S-box prior to the XOR operation. Upon decoding, this operation is reversed. Although the initial permutation substition in DES was performed to improve the runtime of DES on 8-bit machines, the reason for this substitution is unclear to me. It doesn't appear to me to improve either the runtime or the strength of CSS -- but I could be wrong.

The whole process is shown below and can be used for encoding or decoding:



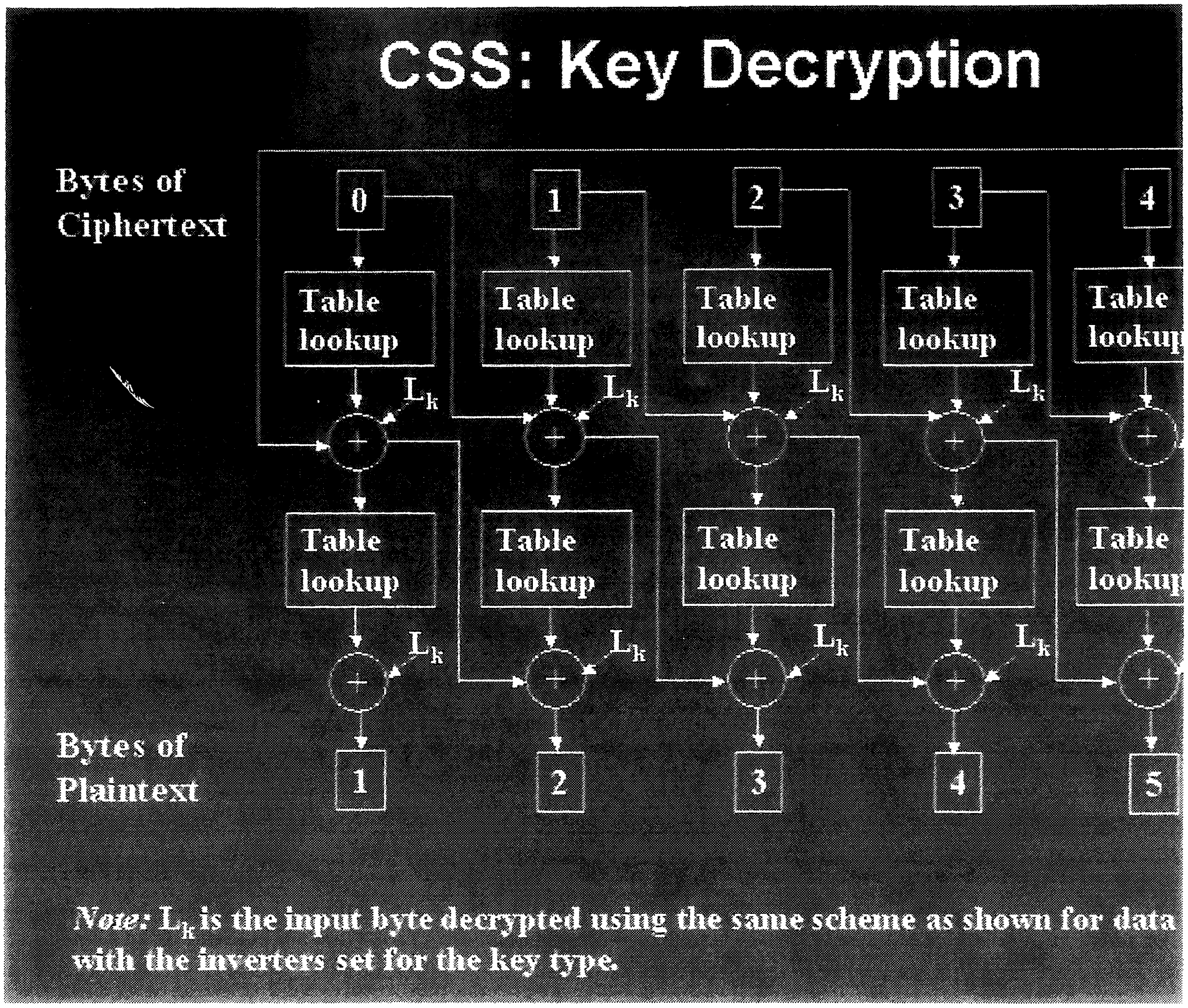
CSS: Data Decryption

- Sector LFSR-17 is seeded with bytes 0 and 1 of the title key XORed with byte 81 of the sector header. A 1 is injected at bit 4, shifting everything right by one
- LFSR-25 is seeded with bytes 2, 3, and 4 of title key XORed with bytes 82, 83, of the sector header. A 1 is injected at bit 4, shifting everything right by one bit
- The output of LFSR-17 is bit-wise inverted before adding to LFSR-25.
- Much as with DES, a table-based substitution is performed on the input data.

Key Encryption

In addition to the encryption described above, CSS goes through an additional two-step mangling operation in the case of keys, including the title key and session key. This process is shown in the slide below.

If we view this slide in terms of columns, each column represents one byte of the key. L_k is the output of the encryption step for the byte represented by the column. The output of the first stage of the mangling function feeds the input of the second stage. The first and second stage are otherwise identical, except for the fact that the output of the 4th byte of the second stage does not wrap around and feed the XOR in the first column.



CSS: Key Decryption

Note: L_k is the input byte decrypted using the same scheme as shown for data with the inverters set for the key type.

Mutual Authentication

Before the DVD player will begin to send data over the bus to the host, it first goth through a form of weak mutual authentication with the host. In the process, it negotiates a key for use in encrypting the data in transit over the bus. This encrpytion is necessary because it would otherwise be possible to snoop the plaintext data right off of the bus, rendering the prior encryption virtually useless. The key that is negotiated is known as the *session key* or *bus key*.

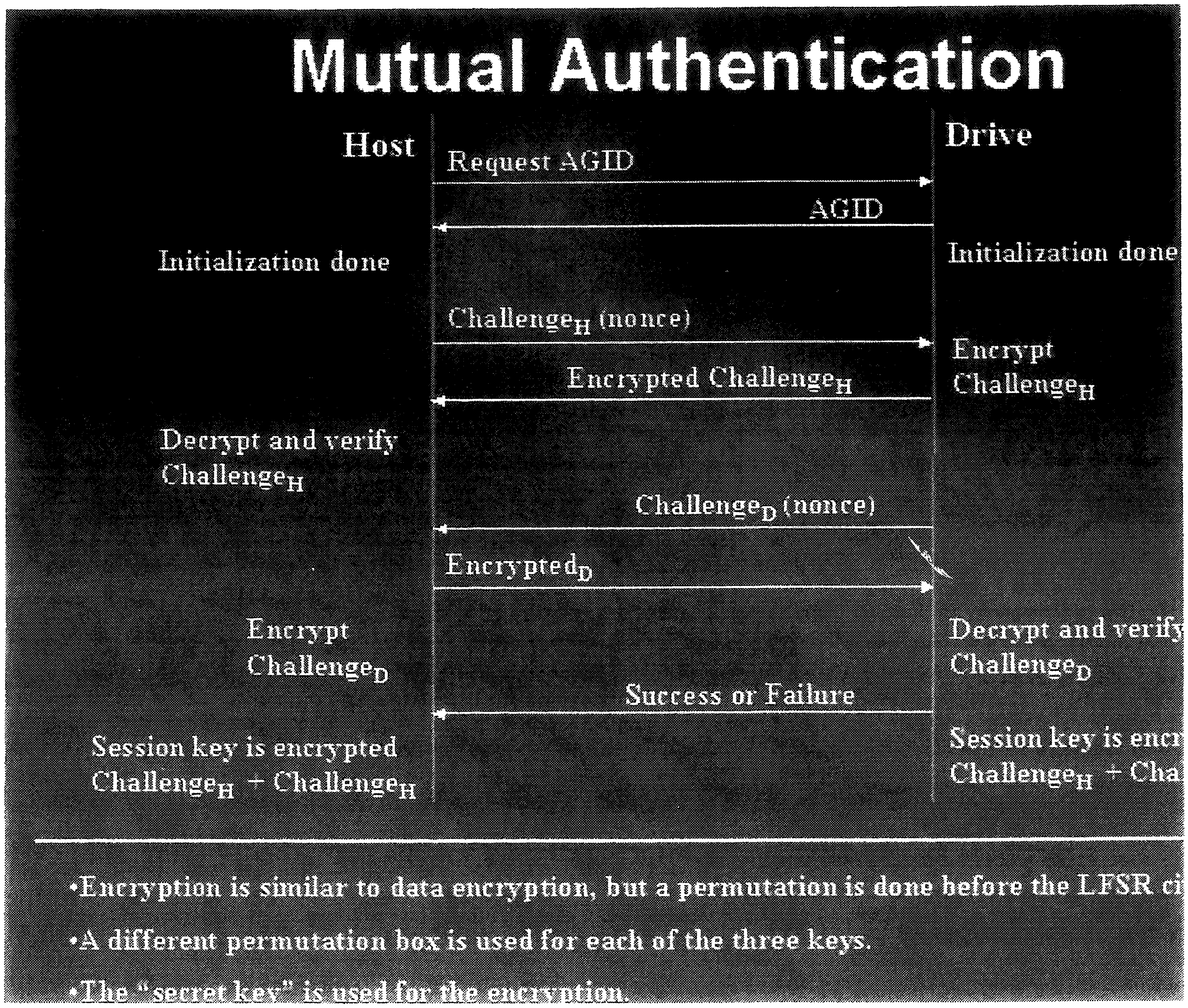
The negotiation begins when the host requests an *Authentication Grant ID (AGID)* from the drive. This ID is much like a session ID or a thread ID. It gives a name to this particular negotiation.

The next thing that happens is the host generates an arbitrary stream of bytes called a *nonce* or *challenge* and sends it to the drive. The drive then encrpyts this stream of bytes and sends them back to the host. The host then decrypts the byte stream and ensures that it is correct. It assumes that the drive is authentic, because it knew the correct secret and algorithm to encode the nonce.

The host performs exactly ther same operation. It generates a nonce, encrpyts it, and sends it to the host. The host in turn encrypts the nonce and sends it back to the drive. The drive then decrypts the nonce and makes sure that it is in fact correct. At this point, both the host and the drive trust each other. This seems to be a fairly weak authentication scheme, because it is based on a secret private key. But this key really can't be all that secret, since it is presumedly in the firmware inside of every DVD player and drive.

Regardless, both the host and the drive now know each other's nonces. Each then takes the two nonces, combines them, and encrypts them as described earlier. The result is the bus key, a.k.a. session key. This key is used to encrpyt all data sent between the drive and the player. Since only the player and the host know the nonces which change every session, only the player and the host can generate the key needed to decrpyt the data.

The only other important note is that, during encrpytion and decryption, a different substitution table is used to perform the initial substition for each of the keys.



Mutual Authentication

Host

Drive

Request AGID

AGID

Initialization done

Initialization done

Challenge$_H$ (nonce)

Encrypt Challenge$_H$

Encrypted Challenge$_H$

Decrypt and verify Challenge$_H$

Challenge$_D$ (nonce)

Encrypted$_D$

Encrypt Challenge$_D$

Decrypt and verify Challenge$_D$

Success or Failure

Session key is encrypted Challenge$_H$ + Challenge$_H$

Session key is encr Challenge$_H$ + Cha

- Encryption is similar to data encryption, but a permutation is done before the LFSR ci
- A different permutation box is used for each of the three keys.
- The "secret key" is used for the encryption.

Weakness #1: Brute Force

Now that we've discussed the CSS algorithm, let's see what we can learn from its weaknesses.

The first thing to note is that the key is only 40 bits long. This isn't a terrbily long key -- it is 16 bits narrower than the DES key. As we discussed last class, even 56 bits can fall somewhat quickly to a prute force attack. The 40 bit length was likely selected to satisfy U.S. export regulations -- but that came at a price.

Weakness #2: 6 Bytes of LFSR Output

The second attack that we are going to talk about requires 6 bytes of LFSR output. It isn't a terribly useful attack, since we don't usually happen to have six bytes hanging around, but it is interesting to talk about, since it provides a 2^{16} attack on the encryption algorithm. In other words, it allows us to crack the whole 40-bit key, if we have 6 bytes of output and crack the 16-bit (plus 1) register by brute force.

Here's how it works:

1. Take a guess at the initial state of LFSR-17.
2. Clock out 4 bytes.
3. Use those 4 bytes to determine the corresponding 4 bytes of output from LFSR-25. This isn't hard, since the two are added -- just subtract.
4. Use the LFSR-25 output to determine LFSR-25's state.
5. Clock out 2 bytes on both LFSRs.
6. Verify these two bytes. Celebrate or guess again.

Weakness #2: 5 Bytes of LFSR Output

Another attack is possible in 2 time, if we know only 5 bytes of output. As you'll see soon, this is a much more practical weakness, because there is yet another weakness that can give us 5 bytes.

1. Guess the initial state of LFSR-17
2. Clock out 3 bytes
3. Determine the corresponding output bytes from LFSR-25
4. This reveals all but the highest-order bit of LFSR-25
5. Try both possibilities:
   1. Clock back 3 bytes
   2. Select the setting where bit 4 is 1 (remember this is the initial case).
   3. It is possible that both satisfy this -- try both.
6. Verify as before

Weakness #3: Mangled Output

This attack can recover 5 bytes of the output of the LFSRs, given both the ciphertext and the plaintext. This 5 bytes can then be used as the 5 output bytes needed for the attack above. Recall the mangling function we talked about earlier. This attack is based on taking a guess and reversing that function.

1. Guess Lk_4
2. Work backward and verify input byte
3. This is a 2^8 attack.
4. Repeat for all 5 bytes -- this gives you the 5 bytes of known output for prior weakness.

Weakness #5: Attacking the Disk Key Hash

There is also a known attack that can recover the disk key from the disk key has in 2^{25} time. This attack is a bit complex, so we won't discuss it. The important observation is that the existence of a 2^{25} attack demonstrates a weakness in the algorithm -- that is a long way from 2.

References

- Axboe, Jens, dvd-2.2.13-5 Linux patch, 1999.
- Fawcus, D. and Roberts, Mark, css-auth package, December, 1999.
- Schneier, Bruce, Applied Cryptography, 2ed, Wiley, 1996, p. 372-379.

- Stevenson, Frank A., "Cryptanalysis of Content Scrambling System", 8 Nov. 1999, as updated 13 Nov. 1999.

Please note:

You should be aware that, in light of a recent federal circuit court decision, it is probably unlawful for you to obtain the the first two sources. To the best of my non-expert and incomplete knowledge, the fourth source has not yet been subject to judicial review in the United States.

These works are cited to "give credit where credit is due". This citation should be viewed as proper attribution **not** suggested reading.

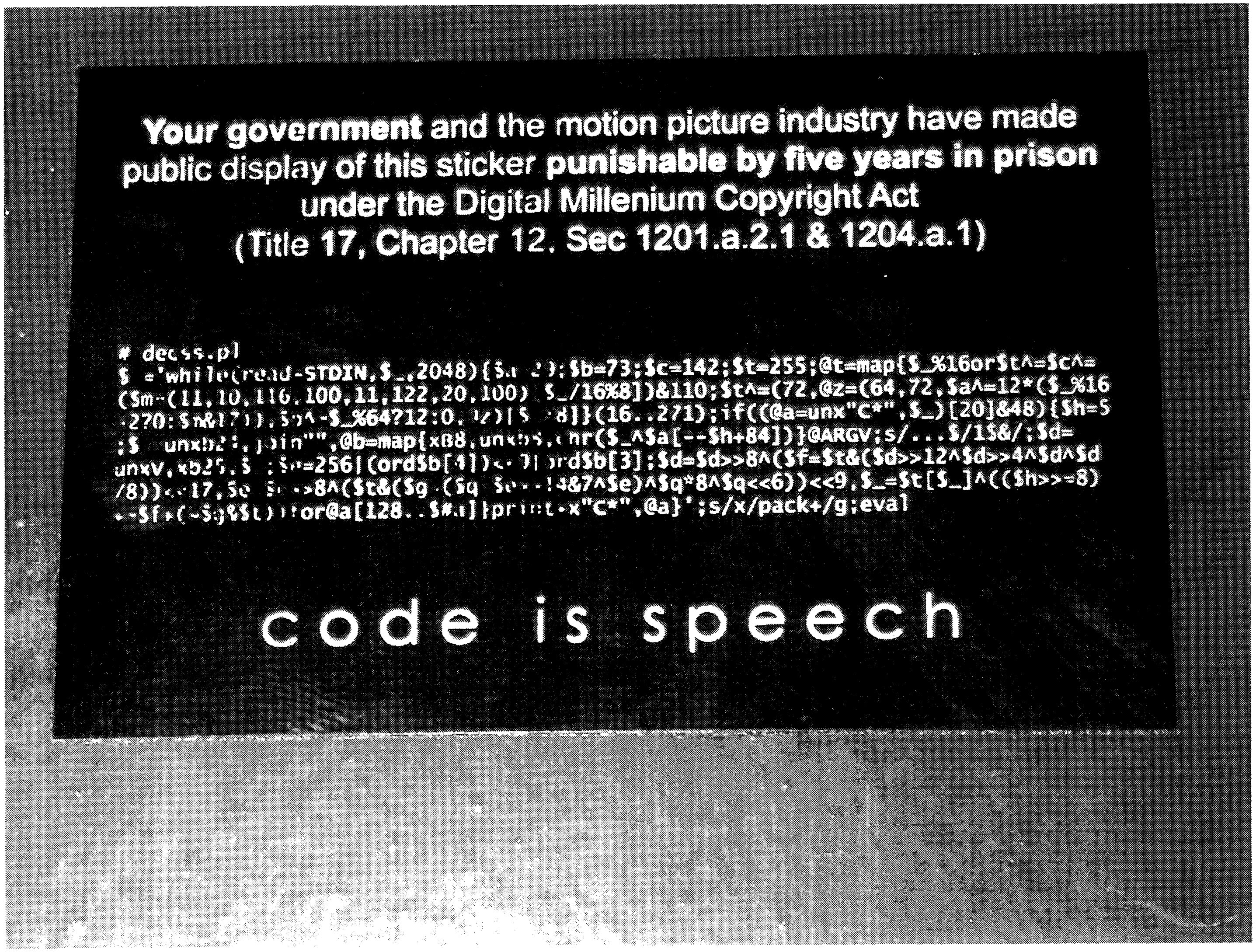
It is my understanding that the recent decision did not incriminate presentations of CSS, such as this one, in detail and form insufficient to constitute a working implementation. But, case law in this area is underdeveloped. As the meaning of the law is further exposed, we (you and I) may find ourselves unable to lawfully distribute or communicate this presentation or its content.

Another note: Take legal advice from a licensed attorney, not from me.

Ex. B to Ex. 4

EXHIBIT B TO THE KESDEN DECLARATION IS FILED WITH THE "HIGHLY CONFIDENTIAL" EXHIBITS

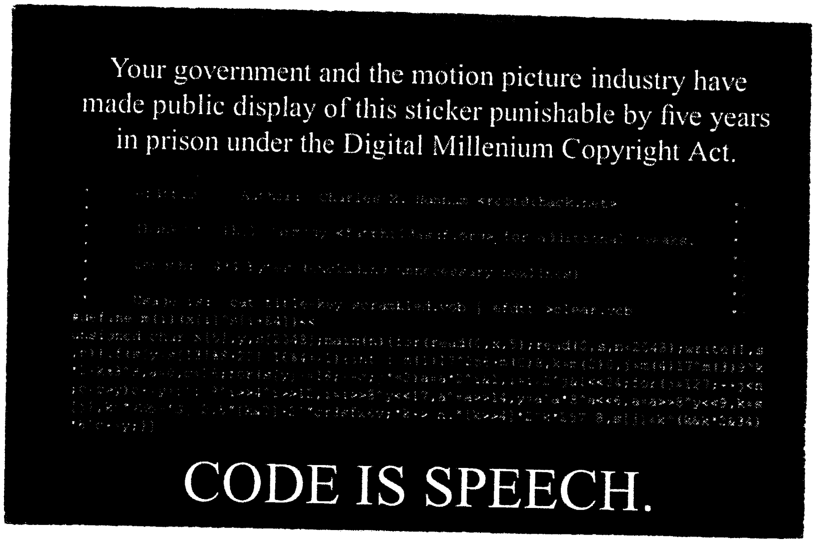
Ex. C to Ex. 4



Your government and the motion picture industry have made public display of this sticker **punishable by five years in prison** under the Digital Millenium Copyright Act (Title **17**, **Chapter** 12, **Sec 1201.a.2.1 & 1204.a.1**)

```
# decss.pl
$_='while(read+STDIN,$_,2048){$a=29;$b=73;$c=142;$t=255;@t=map{$_%16or$t^=$c^=
($m=(11,10,116,100,11,122,20,100)[$_/16%8])&110;$t^=(72,@z=(64,72,$a^=12*($_%16
-2?0:$m&17)),$b^=$_%64?12:0,@z)[$_%8]}(16..271);if((@a=unx"C*",$_)[20]&48){$h=5
;$_=unxb24,join"",@b=map{xB8,unxb8,chr($_^$a[--$h+84])}@ARGV;s/...$/1$&/;$d=
unxV,xb25,$_;$e=256|(ord$b[4])<<9|ord$b[3];$d=$d>>8^($f=$t&($d>>12^$d>>4^$d^$d
/8))<<17,$e=$e>>8^($t&($g=($q=$e>>14&7^$e)^$q*8^$q<<6))<<9,$_=$t[$_]^(($h>>=8)
+=$f+(~$g&$t))for@a[128..$#a]}print+x"C*",@a}';s/x/pack+/g;eval
```

code is speech



Your government and the motion picture industry have made public display of this sticker punishable by five years in prison under the Digital Millenium Copyright Act.

```
/*  efdtt.c     Author:  Charles M. Hannum <root@ihack.net>                 */
/*                                                                          */
/*  Thanks to Phil Carmody <fatphil@asimov.net> for additional tweaks.      */
/*                                                                          */
/*  Length:  434 bytes (excluding unnecessary newlines)                     */
/*                                                                          */
/*  Usage is:  cat title-key scrambled.vob | efdtt >clear.vob               */
#define m(i)(x[i]^s[i+84])<<
unsigned char x[5],y,s[2048];main(n){for(read(0,x,5);read(0,s,n=2048);write(1,s
,n))if(s[y=s[13]%8+20]/16%4==1){int i=m(1)17^256+m(0)8,k=m(2)0,j=m(4)17^m(3)9^k
*2-k%8^8,a=0,c=26;for(s[y]-=16;--c;j*=2)a=a*2^i&1,i=i/2^j&1<<24;for(j=127;++j<n
;c=c>y)c+=y=i^i/8^i>>4^i>>12,i=i>>8^y<<17,a^=a>>14,y=a^a*8^a<<6,a=a>>8^y<<9,k=s
[j],k="7Wo~'G_\216"[k&7]+2^"cr3sfw6v;*k+>/n."[k>>4]*2^k*257/8,s[j]=k^(k&k*2&34)
*6^c+~y;}}
```

CODE IS SPEECH.

Ex. 5

RICHARD R. WIEBE (SBN 121156)
2140 Ninth Avenue
San Francisco, CA 94116
Telephone: (415) 505-8793
Facsimile: (240) 282-7297

THOMAS E. MOORE III (SBN 115107)
TOMLINSON ZISKO MOROSOLI & MASER LLP
200 Page Mill Road, Second Floor
Palo Alto, CA 94306
Telephone: (650) 325-8666
Facsimile: (650) 324-1808

ALLONN E. LEVY (SBN 187251)
HS LAW GROUP
210 N. Fourth St., Second Floor
San Jose, CA 95112
Telephone: (408) 295-7034
Facsimile: (408) 295-5799

ROBIN D. GROSS (SBN 200701)
ELECTRONIC FRONTIER FOUNDATION
454 Shotwell Street
San Francisco CA 94110
Telephone: (415) 436-9333
Facsimile: (415) 436-9993

Attorneys for Defendant ANDREW BUNNER

SUPERIOR COURT OF THE STATE OF CALIFORNIA

COUNTY OF SANTA CLARA

DVD COPY CONTROL ASSOCIATION, INC.,
Plaintiff,
v.

ANDREW THOMAS MCGLAUGHLIN; ANDREW BUNNER; et al.,
Defendants.

Case No CV - 786804

DECLARATION OF COMPUTER SCIENTIST ROLAND PARVIAINEN

IN SUPPPORT OF DEFENDANT ANDREW BUNNER'S MOTION FOR SUMMARY JUDGMENT

I, Roland Parviainen, declare:

1. I am an instructor in the Computer Science and Electrical Engineering Department of Luleå University of Technology in Luleå, Sweden. I received my M.S. degree in Computer Science in 1999 from Luleå University and since that time have been a graduate student in the Ph.D. program in the Computer Science and Electrical Engineering Department at the University.
2. I have taught the Computer Security course three times for the Computer Science and Electrical Engineering Department of Luleå University.
3. I most recently taught the course in Computer Security in the Spring Term of 2001. As part of my course, I taught my students how the Content Scrambling System ("CSS") encrypts and decrypts DVD movie disks.
4. Information about CSS is freely available within the computer science community. In preparing to teach my Computer Security course, I had no difficulty in obtaining from sources on the Internet the information required to understand the CSS algorithms and keys and to understand how those algorithms and keys are used to encrypt and decrypt DVD movies. These sources include Frank A. Stevenson's paper "Cryptanalysis of Contents Scrambling System" as well as source code for various DVD decryption programs. These decryption programs include "DeCSS," a program that decrypts DVD movie disks by implementing the CSS decryption algorithms and keys, as well as other programs that are functionally equivalent but instead exploit weaknesses in the CSS algorithms to decrypt the movie data without replicating the CSS decryption process. The source code available on the Internet for these programs reveals how the CSS algorithms and keys work.

5. I use CSS in my Computer Security course as an example of how not to design an encryption system. CSS is an inherently weak encryption system that can easily be attacked. It has a weak algorithm, relatively short (40-bit) keys that are vulnerable to brute-force attack, and in the case of software DVD players the player key is stored in the computer's memory.

I, ROLAND PARVIAINEN, declare under penalty of perjury under the laws of the State of California that the foregoing is true and correct.

Dated: 10/26/2001

Roland Parviainen

Ex. 6

RICHARD R. WIEBE (SBN 121156)
425 California Street, Suite 2025
San Francisco, CA 94104
Telephone: (415) 433-3200
Facsimile: (415) 433-6382

THOMAS E. MOORE III (SBN 115107)
TOMLINSON ZISKO MOROSOLI & MASER LLP
200 Page Mill Road, Second Floor
Palo Alto, CA 94306
Telephone: (650) 325-8666
Facsimile: (650) 324-1808

ALLONN E. LEVY (SBN 187251)
HS LAW GROUP
210 N. Fourth St., Second Floor
San Jose, CA 95112
Telephone: (408) 295-7034
Facsimile: (408) 295-5799

ROBIN D. GROSS (SBN 200701)
ELECTRONIC FRONTIER FOUNDATION
454 Shotwell Street
San Francisco CA 94110
Telephone: (415) 436-9333
Facsimile: (415) 436-9993

Attorneys for Defendant ANDREW BUNNER

SUPERIOR COURT OF THE STATE OF CALIFORNIA

COUNTY OF SANTA CLARA

DVD COPY CONTROL ASSOCIATION, INC., Plaintiff, v. ANDREW THOMAS MCLAUGHLIN; ANDREW BUNNER; et al., Defendants.	Case No. CV - 786804 **DECLARATION OF DEFENDANT ANDREW BUNNER** **IN SUPPPORT OF HIS MOTION FOR SUMMARY JUDGMENT**

I, ANDREW BUNNER, declare:

1. I am one of the defendants in this action.
2. I had no involvement in or first-hand knowledge of the creation or programming of DeCSS or any other DVD descrambling program.
3. I had no involvement in or first-hand knowledge of whatever reverse engineering or technical analysis of CSS may have occurred in connection with the creation of DeCSS or any other DVD descrambling program.
4. I had no involvement in the original publication of DeCSS or any other DVD descrambling program on the Internet.
5. I first learned of DeCSS after CSS had been reverse engineered and after DeCSS had been created and published in October 1999 on the Internet.
6. After the creation and publication on the Internet of DeCSS, I then downloaded a copy of DeCSS from an unrestricted, publicly available web site on the Internet and placed it on my personal web site.
7. In December 1999, before being served with the summons and complaint in this action, I spoke by telephone by an attorney for plaintiff DVD Copy Control Association, Inc. ("DVD CCA"). I immediately removed DeCSS from my web site server during my telephone conversation with DVD CCA's attorney. I have not disclosed or distributed DeCSS or any other DVD descrambling program since that time and I have obeyed the Court's preliminary injunction.

I, ANDREW BUNNER, declare under penalty of perjury under the laws of the State of California that the foregoing is true and correct.

Dated: 11/15/2001

Andrew Bunner

Ex. 7

WEIL, GOTSHAL & MANGES LLP
JARED BOBROW (State Bar No. 133712)
CHRISTOPHER J. COX (State Bar No. 151650)
2882 Sand Hill Road, Suite 280
Menlo Park, California 94025
Telephone: (650) 926-6200
Facsimile: (650) 854-3713

WEIL, GOTSHAL & MANGES LLP
ROBERT G. SUGARMAN
EDWARD J. BURKE (State Bar No. 103414)
JONATHAN S. SHAPIRO
RICHARD A. SIMON
767 Fifth Avenue
New York, New York 10153
Telephone: (212) 310-8000
Facsimile: (212) 310-8007

Attorneys for Plaintiff
DVD COPY CONTROL ASSOCIATION, INC.

SUPERIOR COURT OF THE STATE OF CALIFORNIA
COUNTY OF SANTA CLARA

DVD COPY CONTROL ASSOCIATION, INC., a not-for-profit trade association,

Plaintiff,

v.

ANDREW THOMAS MCLAUGHLIN, an individual, et al.,

Defendants.

Case No. CV 786804

PLAINTIFF'S ANSWERS AND OBJECTIONS TO DEFENDANT ANDREW BUNNER'S FIRST SET OF SPECIAL INTERROGATORIES

RESPONDING PARTY: Plaintiff, DVD COPY CONTROL ASSOCIATION, INC.
SET NUMBER: FIRST SET OF SPECIAL INTERROGATORIES
PROPOUNDING PARTY: Defendant, ANDREW BUNNER

Pursuant to Section 2033(f) of the Cal. Rules of Civ. Proc., Plaintiff, DVD COPY CONTROL ASSOCIATION, INC. ("DVD CCA"), by and through its attorneys, Weil, Gotshal & Manges LLP, hereby provides the following answers and objections to Defendant ANDREW BUNNER's First Set of Special Interrogatories, dated March 8, 2000 (the "Interrogatories"):

PRELIMINARY STATEMENT

Plaintiff has not completed its investigation of the facts relating to this case, has not completed discovery, and has not commenced its preparation for trial. As such, these Answers are given without prejudice to Plaintiff's right to produce evidence of any subsequently discovered fact or facts, and to otherwise assert factual and legal contentions as additional facts are ascertained, further analyses are made, and ongoing legal research is completed. Thus, Plaintiff reserves the right, without incurring the obligation, to amend or supplement these answers as investigation and discovery continue.

Plaintiff reserves the right to make future motions and objections relating to the Interrogatories at any subsequent state of this action, including, but not limited to, the right to the use of any information on any ground in any proceeding in any action (including at trial) and the right to object on any ground at any time to any demand for a further answer to these or to any other Interrogatories.

Plaintiff reserves all objections as to the admissibility of evidence, at trial or otherwise, of information produced pursuant to these Answers, including, without limitation, objections on grounds of relevance and materiality. These Answers are made without waiver of, or prejudice to, these or any additional objections Plaintiff may make. All such objections are hereby expressly preserved as is the right to move for a protective order.

Hundreds of companies – including Plaintiff and its licensees – depend on the security of the DVD format as an integral part of their business, including but not limited to hardware and software manufacturers of DVD players, DVD manufacturers, DVD content providers and DVD resellers. These companies have invested millions of dollars in the development and production of DVD-oriented products. If the technology upon which these companies rely no longer viably protects the security of the goods they produce and their contents, those companies – including Plaintiff – will generate less revenue and will eventually go out of business.

SPECIAL INTERROGATORY NO. 25

Provide the name, address and phone number for any individual or entity that is a "DVD Copy Control Association predecessor-in-interest" as defined in your complaint.

ANSWER TO SPECIAL INTERROGATORY NO. 25

Plaintiff states that the predecessors-in-interest of DVD CCA, as that term is used in the Complaint filed in this action, are Matsushita Electric Industrial Co., Ltd. ("MEI"), 1006 Kadoma, Osaka 571-8501, Japan, Tel. 81-6-6908-1121; Toshiba Corporation ("Toshiba"), 1-1, Shibaura 1-chome, Minato-ku, Tokyo 105-8001, Japan, Tel. 81-3-3457-2096; and CSS Interim Licensing Organization ("CSS ILO"), 2-15, Matsuba-cho, Kadoma, Osaka 571-8509, Japan. Tel. 81-6-6905-4155.

SPECIAL INTERROGATORY NO. 29

State all facts in support of the contention in your complaint that the source code of the program named "DeCSS" was first posted on the Internet by Jon Johansen.

ANSWER TO SPECIAL INTERROGATORY NO. 29

Plaintiff objects to this Interrogatory to the extent that it seeks information that is unavailable to Plaintiff or that cannot be readily obtained without undue labor and expense. Plaintiff further objects to this Interrogatory on the grounds that it is overly broad and unduly burdensome, including, without limitation, that it seeks to require Plaintiff to provide information other than that which may be obtained through a reasonable diligent search of its files and records.

ripp the two sides independent. I then end up with two 4.5GB files. Is there any way I can multiplex theese files afterwards? I really like to have one avi file, instead of two."

A post by JsTwNaHvFn <jstwnahvfn@aol.com> to the Usenet newsgroup alt.hacking on July 10, 2000 states: "Used decss to copy green mile movie, can't get sound is their a reason an a way to copy sucessfully an even a better copying program than DeCss, an where can i pick up a copy, sure appericate any an all info....feel free to e-mail the info...."

A post by Stone <mstone1@nyc.rr.com> to the Usenet newsgroup alt.toys.gi-joe on July 6, 2000 states: "Anyone else having trouble with the G.I. Joe the Movie DVD that just came out last week?? I have never had a problem playing any movies in my computer DVD drive and I have over 50 movies! This crappy disc from Kid Rhino just goes black after the FBI warning and Kid Rhino logo appear. I've even tried playing it in software playback mode, ripped it with DeCSS, and still the crap is all scrambled.

Plaintiff further states that in the Shamos Dec., Professor Michael I. Shamos presented evidence of the process by which motion pictures can be and have been pirated using DeCSS.

SPECIAL INTERROGATORY NO. 38

Define the term "pirating" as used in your complaint.

ANSWER TO SPECIAL INTERROGATORY NO. 38

Plaintiff states that the term "pirating" refers to the actual or attempted illegal or unauthorized use, copying, transfer, reproduction, viewing, sale, rental or distribution of material owned by another person, without such person's permission, or to actions which may aid or assist in such activities. "Pirating" therefore includes the copy and transfer of DVD movies to others, i.e. "swapping" copyrighted works, as described in the Shamos Dec.

SPECIAL INTERROGATORY NO. 39

State all facts in support of the contention in your complaint that the Algorithms contained within CSS are proprietary property and trade secrets.

ANSWER TO SPECIAL INTERROGATORY NO. 39

Plaintiff objects to this Interrogatory to the extent that it seeks information that is unavailable to Plaintiff or that cannot be readily obtained without undue labor and expense. Plaintiff further objects to this Interrogatory on the grounds that it is overly broad and unduly burdensome, including, without limitation, that it seeks to require Plaintiff to provide information other than that which may be obtained through a reasonable diligent search of its files and records.

Plaintiff objects to this Interrogatory to the extent that it is vague, insufficiently specific, uses undefined terms capable of more than one interpretation, or requires Plaintiff to speculate as to the specific information demanded.

Plaintiff objects to this Interrogatory to the extent that it seeks information protected by various privileges or immunities, including the attorney-client privilege, the work product doctrine or any other applicable privilege.

Plaintiff objects to this Interrogatory to the extent that it seeks to require Plaintiff to produce information not in its possession, custody or control.

Subject to these objections, Plaintiff states that the CSS system is proprietary. Toshiba and MEI, DVD CCA's predecessors, engineered the technology and hold certain intellectual property rights with respect to it. DVD CCA and its predecessors, as well as the members of the motion picture, information technology and electronics industries, have invested substantial amounts of money and resources in the development of the DVD application and the necessary safeguards, such as the CSS licensing mechanism, to protect the DVD copyrighted content. The CSS license and, in particular, the manner in which it protects DVD technology, is of great commercial importance to the motion picture, information technology and computer electronics industries. These intellectual property rights, including the algorithms contained within CSS, are currently licensed by DVD CCA as the sole duly authorized licensing entity for the CSS technology.

Beginning on or about October 31, 1996, DVD CCA's predecessor CSS ILO began licensing CSS technology pursuant to an agreement that later became the Amended And Restated CSS Interim License Agreement, including the related CSS Procedural Amended And

Restated Technical Specifications (collectively, the "CSS Agreement"). The CSS Agreement sets forth the terms and conditions under which the CSS licensing entity (currently DVD CCA) would grant licenses to, among others, manufacturers of DVD players or DVD drives and related hardware and software. Licensees were granted the right to use the security system on DVD products and agreed to safeguard the CSS technology from public disclosure.

The CSS Agreement gives the licensees the right to use the technology and may provide the necessary algorithms, decscrambling technology and/or "master keys" to do so, depending on the licensee's category. The proprietary technology is not accessible to unlicensed third parties because it is incorporated in hardware devices – chips – or made tamper resistant if distributed in the form of actual software. Both forms of distribution are such that the proprietary technology cannot be viewed by non-licensees. When the DVD system was created, approximately 400 unique "master keys" were predesignated, and were to be assigned over time to licensees in certain license categories. Each DVD contains, in a part of the disc not normally read by the player device, a sector of the "Secured Disc Key data." The system will not operate unless the key contained in the licensee's decryption module (a chip or software program) matches one of the "master keys" and is used to recover the Disc Key from the "Secured Disc Key data" on the DVD.

The CSS Agreement requires licensees to maintain the confidentiality of certain defined pieces of information, such as the algorithms and "master keys" and, as such, licensees are subject to a very stringent set of rules to ensure the maintenance of confidentiality within the group of licensees.

Section 5.2 of the CSS License Agreement contains the confidentiality restrictions imposed on the licensees to protect the proprietary nature of the CSS technology. The CSS License Agreement requires licensees to maintain the confidentiality of certain defined pieces of information, such as the algorithms and "master keys." As such, licensees are subject to a stringent set of rules to ensure the maintenance of confidentiality within the group of licensees.

Among the safeguards taken is the requirement that only those licensees that absolutely need to know a particular algorithm and/or key are provided with such information.

For example, a manufacturer of semiconductor chips for descrambling CSS content in stand-alone DVD players is provided with information necessary for manufacturing such chips, but not with information concerning the scrambling process itself or the authentication between DVD drives and the descrambling module used for computer-based implementations. Companies that merely assemble parts and components produced by others may be required to be licensees in order to purchase such parts and components, but these companies are not provided with the proprietary CSS information at issue.

The CSS License Agreement mandates that licensees provide only the proprietary CSS technology at issue to the strictest minimum number of a licensee's employees who require access to the information, beginning with only three employees and expanding beyond three only upon notification to the licensor of the names of the additional employees. Licensees who violate these requirements are subject to liquidated damages in the amount of $1 million per violation (with a cap based on profits made from the sale of licensed products).

Licensees implementing authentication and descrambling functions in software are required to do so only in a manner that obscures the proprietary CSS technology at issue, so as to effectively frustrate anyone seeking to obtain such proprietary information. The CSS Agreement requires such implementations to be designed with a degree of robustness that cannot be defeated or circumvented without employing a high degree of technical skill. Failure to abide by these operating restrictions can subject the licensee to injunctions prohibiting the sale of the product in which the failure occurs, through actions brought either by the licensor or by third party beneficiary content owners (motion picture studios that are licensees under the CSS License Agreement and have made copyrighted content available on DVDs encrypted using CSS technology).

SPECIAL INTERROGATORY NO. 40

State all facts in support of the contention in your complaint that the CSS master keys are proprietary property and trade secrets.

ANSWER TO SPECIAL INTERROGATORY NO. 40

Plaintiff objects to this Interrogatory to the extent that it seeks to require Plaintiff to produce information not in its possession, custody or control.

Dated: August 10, 2000

WEIL, GOTSHAL & MANGES LLP

By: ______________________
EDWARD J. BURKE
(State Bar. No. 103414)

Attorneys for Plaintiff
DVD Copy Control Association, Inc.

CORPORATE VERIFICATION

I, John J. Hoy, declare as follows:

I am President of DVD Copy Control Association, Inc. ("DVD CCA") and, as such, I am authorized to make this Verification for and on behalf of DVD CCA, a party to this action, and I make this Verification for that reason. I have read the foregoing PLAINTIFF'S ANSWERS AND OBJECTIONS TO DEFENDANT ANDREW BUNNER'S FIRST SET OF SPECIAL INTERROGATORIES, and I am informed and believe, and on that ground allege, that the matters stated in the foregoing document are true.

I declare under penalty of perjury under the laws of the State of California that the foregoing is true and correct and that this Verification was executed this 27th day of August, 2000, at Morgan Hill, California.

DVD COPY CONTROL ASSOCIATION, INC,

By:______________________________________
John J. Hoy, President

EXHIBIT B

Court of Appeal, Sixth Appellate District - No. H021961
S100809

IN THE SUPREME COURT OF CALIFORNIA

En Banc

MATTHEW PAVLOVICH, Petitioner,

v.

THE SUPERIOR COURT OF SANTA CLARA COUNTY, Respondent;

DVD COPY CONTROL ASSOCIATION INC., Real Party in Interest.

Real Party in Interest's " Motion To Extend Finality of the Decision" filed on December 17, 2002, is DENIED.

Baxter, J., is of the opinion the motion should be granted.

Chief Justice