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: ___________________   _________________________

David A. Wagner