SOFTWARE AS SPEECH

Dan L. Burk

I'm delighted and very pleased to be here this morning, and want to thank the members of the law journal for inviting all of us to participate in this program. I think you'll discover as you look at your program that, as Mark Alexander has just shown and as I'm going to show you in a moment, the great genius of this program is that the morning session consists of law school professors who will leave you all with unanswered questions. The afternoon program is composed of more practical people who will actually give you some answers. I'll leave you with probably more questions here than I answer because, as the title of my talk indicates, I want to talk about the intellectual property implications of software as expression.

Much of what I do interacting with the First Amendment is as the flip side of the law version of my interest in intellectual property, and that interest was piqued a little while back from a decision in California, *Bernstein v. United States*,[1] that deals with cryptography and the implications of cryptographic software, and in particular a claim made in the court there that computer software is speech that is fully protected by the First Amendment and so deserves the full range of rights and protections against government prior restraint. I want to explore what that means, because the court held that software is speech, and did so by drawing on the law of copyright as applied to software. That application of copyright jurisprudence to First Amendment jurisprudence has some disturbing implications, as I hope to show this morning, and I will argue that the road we've gone down with regard to copyright in software is a road that we don't want to go down with regard to free speech.

Public key cryptography. I'll begin by talking briefly about cryptography since that was the subject of the *Bernstein* suit. I won't spend a lot of time describing cryptography to you, because I know Professor Nuara, my colleague here, is a going to talk about cryptography this afternoon, and I'm sure he'll give you a good rundown on what it is and the free speech implications surrounding it. For the purpose of our discussion this morning, let me simply say that cryptography comprises methods of encoding communications that take advantage of some sort of principles of mathematics, particularly in character-

[1]922 F.Supp. 1426 (N.D. Cal. 1997); 974 F.Supp. 1288 (N.D. Cal. 1997).

istics of very, very large prime numbers. In some cases, we will have symmetric cryptography, where the key used to encrypt a message is the same one used to decrypt the message—exactly like the kind of encryption you would do with your Captain Midnight decoder ring.

Alternatively, we can have asymmetric encryption, where the mathematics of large numbers can be used to create pairs of keys, pairs of methods for encoding messages. We can break those up so that there is a public key which you can publish and give to the whole world and a private key that you keep to yourself. The private key, when it's used to encode something, that message can only be decoded by a public key. And vice versa: things encoded by a public key can only be decoded by the private key. Such keys, and the processes of encryption and decryption, can be performed by computers using software designed to carry out those functions. And so these pairs of keys enable a number of very important kinds of transactions on the Internet.[2]

First of all, encryption of this sort enables secure communications. This morning, Ed Cavazos talked about some types of discussions that you might be concerned about having, even on the Internet. If you're a man, you may not want the whole world to know about your fascination for women's shoes, for example. When you're ordering those pumps, you might want to encode that communication so that not everybody can read it. By the same token, you might want to be secure in transmitting financial information like your credit card number. You might also want to use this technology to enable digital signatures so you know for certain who a particular message came from. Or you might want to enable other types of technologies that would facilitate online transactions and the electronic commerce.

So this whole question of cryptography is important to free speech and to commercial transactions on the Internet. Now, it also makes the government very unhappy, because this type of encryption, which uses these very large prime numbers, is for all practical purposes, more or less unbreakable. Once encoded in this manner, the message can only be decoded with the proper key. With enough computing power you might be able to decode the message without having the key, but in many cases it's estimated that that would take longer than the age of the universe to do that, and the government doesn't want to wait that long to read the message if they feel there's a need to read your message. Consequently, the government is trying to restrict the availability of this technology. The restrictions go back to the cold war, and this technology is listed as a type of a munition, like guns and tanks and airplanes, that we don't want the enemy to get a hold of.

More recently the government has professed to be concerned about criminal

[2]*See generally* A. Michael Froomkin, *Flood Control on the Information Ocean*, 15 J.L. & COM. 395, 449-50 (1996).

activity; child pornographers, terrorists, mother rapers, father stabbers, father rapers, and all the other horrible people who are supposed to be lurking online: all these folks that might be on the Internet using these very private, unreadable, secure channels to communicate nefarious plans. So the government has tried to use export restrictions to keep the stuff from getting into the hands of Godless communists and all these other kinds of evil people who might want to use it.[3] Of course, the mathematics for developing such cryptographic software are well known, so anyone outside the U.S. can write their own program, they just can not buy an American one. And there are at this point no domestic restrictions on sale of the software, so anyone who wants it within the U.S. can get it, too, including criminals. And of course, criminals are not terribly concerned about obeying export restrictions, because they are criminals. As the saying goes, if crypto is outlawed, only outlaws wil have crypto.

Now I have to say in the interest of full disclosure, I think this entire program on the part of the government is misguided and ultimately futile. That's futile, not feudal. Although it may be feudal as well. First, the governmental restrictions appear to be unconstitutional. For reasons that Professor Nuara will tell you this afternoon, it offends the First Amendment; and it probably offends the Fourth Amendment.[4] Second, it's economically harmful. The Germans think our cryptographic export restrictions are great because it leaves them able to essentially control the whole field of the international market for these very valuable types of software, and they wish we would go ahead and do stupid things like this for the indefinite future. Third, it appears to be inconsistent. The information is getting out there into people's hands anyway, and this doesn't appear to be a particularly good way to keep that from happening.

So as a consequence, some folks have decided to challenge these export restrictions, and one particular case that I'm going to focus on was the challenge in California brought by a computer scientist by the name of Bernstein who wanted to be able to go out of the country and take his cryptographic software with him, and talk about it at academic conferences and discuss ideas about cryptography and mathematics with colleagues around the world. It was a challenge on the ITAR[5] and EAR[6] restrictions on export of this type of soft-

[3]*See, e.g.,* 22 U.S.C. § 2778 (1990; 22 C.F.R §§ 120-30 (1994); *see also generally* A. Michael Froomkin, *It Came From Planet Clipper: The Battle Over Cryptographic Key "Escrow,"* 1996 U. CHI. L. FORUM 15 (discussing United States' cryptographic export policies).

[4]*See* A. Michael Froomkin, *The Metaphor is the Key: Cryptography, the Clipper Chip, and the Constitution,* 143 U. PA. L. REV. 709 (1995).

[5]*See* International Traffic in Arms Regulations, 22 C.F.R. §§ 120-130 (1994).

[6]*See* Export Administration Regulations, 15 C.F.R. Pt. 730 et esq. (1997).

ware, and the challenge that was brought by Bernstein was a First Amendment challenge. He claimed that software is expression, that it is speech. Notice that he did not claim that it enables speech, that it's a method of enabling communication, but rather that the software itself, the code is itself, is expression and should be protected as speech. The court agreed with that. The district court in *Bernstein* said accepted that argument that software is speech, and as a consequence, held that the export restrictions are prior restraints and are unconstitutional under the case law we have for the First Amendment.

Now, the court in doing this had to deal with the question of functionality of software. Because the government, of course, viewed the software was not speech. The government classifies the software as a munition like a gun or tank or some sort of machine, rather than as being expressive. The court said, no, it thought that the software is kind of like a piano roll. A piano roll is part of a mechanism, part of a player piano but it plays music, and we know music is expressive and protected by the First Amendment. And if you know anything about intellectual property, particularly about copyright software, when you say piano roll, the hair on your head stands up on end.[7] The analogy is very familiar. In fact, the court went on to say that we know from a copyright statute that software is expression. Congress told us that software is copyrightable, and we know that copyright protects expression, therefore the court concluded that software must be expressive and protected for purposes of the First Amendment.

Now, as I said a moment ago, I'm delighted with that outcome. I do think that this program by the government is a bad way to approach the question of cryptography. But I'm also a little disturbed by the court's approach to the problem, and particularly by the court's direct comparison to the intellectual property protection of software under copyright, because we have to think for a minute about what software is. Ultimately, software consists of a set of instructions to a machine. We talk about source code, which is the funny scribbles you see programmers writing on yellow pads or typing into computers that appear on the screen, the argument made in Bernstein was that other programmers can read Bernstein's source code when he writes a program. So he argued that's communicating to other programmers.

The court in California accepted that source code is expression. Other programmers can read it and understand the ideas that are in it. Ultimately, though, in a computer, the source code has to be turned into object code, which is the version of the program that can be read and executed by a machine. A programmer is probably not writing software just for the fun of it; she's probably writing it to get the machine to do something. And using software as in-

[7]*See* White-Smith v. Apollo Publishing Co., 209 U.S. 1 (1980) (holding that a piano roll was not copyrightable because it was read by a machine rather than by humans).

struction to a machine is sort of an interesting artifact of the choice that we made to have machines that have multiple purposes. We can configure this computer in front of me to do a whole lot of different things using software. Or, instead of doing that, we could have built a bunch of different machines in hardware, hardwiring each of them to do one of the things that this machines does. It seemed better to us for a lot of reasons to allow the machine to do lots of different things, be sort of a multipurpose machine and use software to configures it instead of building dedicated machines. But the point is the hardware can do anything the software does, just perhaps not as conveniently.[8]

In the copyright context, people have talked about what this means for expression. If I were to build a hardwired version of a word processor rather than write a software to configure my desktop computer, if I hard wired a computer, would we consider that to be expression and speech? We could draw an blueprint analogy to blueprints. Source code, scribbles that programmers write, that's sort of like a blueprint and the object code is sort of like what you make out of a blueprint. I can design a car or a running shoe or something on a blueprint and then the factory implements that in three dimensions as a sort of hardware, so maybe that's where we ought to draw the line: between the scribbles and the implementation; the former could be expression, and the latter would not.

That line may not work terribly well, though, as we know from our experience with copyright software. There is not as much separation between source code and object code as there is between a blueprint and a manufactured product; it is as if the blueprint were the car. If you can't draw that line, then you have to ask the question, well, where do we draw the line? If my computer software is speech, is my Honda Civic also speech? An engineer thought very hard about what the design of that automobile ought to be, and eventually it was implemented and produced as a car, so in a sense it communicates his ideas. Or should my Nike running shoes be speech because someone thought hard about the design of that and that was eventually implemented in latex? Ultimately is everything around us speech?

Now, that's not a completely crazy idea. If you talk to people who study technology, they say that is the whole point of the humanities. We think that we can look at an old arrowhead or cathedral or something and it has imbedded in it some of the thoughts and characteristics of the civilization that produced it and we can tell things about people by the artifacts that they produce. We talk about thoughts and values being inscribed in artifacts. That's what archaeology is all about, that's what architecture is about, that's what humanities are about, and in fact it's not a terribly controversial idea that our values and thoughts are

[8]*See* Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2319 (1994).

inscribed in technology; for example, in computers. But we have to ask the question is that really where we want the First Amendment to go—is that the sort of expression that the First Amendment ought to protect.

The reason I ask that question is because, is as I said, we've seen this before. As an intellectual property expert, I have a feeling of deja vu when I read *Bernstein* and the controversy that surrounded it because way back at the dawn of copyright protection of software, we had the same debates. Congress appointed a special commission or CONTU, a group of people that looked at the question how should we protect computer software, and advise Congress as to what intellectual property regime should we use.[9] The majority of the commission said we ought to use copyright to protect software, and did not think it was a problem that software is ultimately a functional utilitarian thing. They felt software is expressive because a programmer writes source code and that expression ought to be protected in copyright. There's one very famous dissent from that report by Commissioner Hersey where he said this is going to really distort the law of copyright because copyright doesn't protect functional things. Ultimately he thought software was functional, but his view did not prevail.

The courts got a hold of the question after Congress enacted copyright legislation to protect software, and they bought the CONTU argument as well. For example, in the very notable *Apple v. Franklin*[10] the defendant argued that machine-readable object code could not be protected by copyright, because copyright does not cover functional articles. The argument was that even if Congress decided to protect source code under copyright, it couldn't possibly have meant to protect object code. The machine readable version acts just like a piece of hardware, but the courts in *Franklin* and in other cases held that Congress, must have intended to protect the object code as well, because it doesn't make sense to protect one and not the other. They're so easily converted back and forth, that if you protected only the source code, infringers would simply go to object code. And then the court also rejected a claim that language or communication has to be directed to humans to be the subject of copyright. Because there are programs running in your computer that you never see that manage how a computer works, and people said surely that's not copyrightable, that's not expression, because it's not something that communicates to people. And the court said it doesn't have to, that's not a requirement.

[9]*See generally* Pamela Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form*, 1984 DUKE L.J. 663 (reviewing and critiquing CONTU deliberations); *but see also generally* Arthur R, Miller, *Copyright Protection For Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 HARV. L. REV. 977 (1993) (defending CONTU recommendations).

[10]714 F.2d 1240 (3rd Cir. 1983).

Even today, the courts are still struggling with this. You can look at even very recent decisions where, as a result of Congress trying to shoehorn software into the copyright box, the courts are still struggling with exactly what are the parameters of protection for copyright.[11] And you can look at appellate decisions that try very hard to distinguish what's functional, what's not functional and how far these protections extend. So we've seen difficulties as a result of copyright law as a result of the decision Congress made.

In addition, we're now in a situation where software is not just copyrightable, it's also patentable. Once it didn't look like that would be the case. Patents cover processes, articles of manufacture and machines and compositions of matter.[12] People argued that because software is a set of instructions to a machine, it is really a process, so the patent law ought to cover it. The Supreme Court initially decided against this, partly because they didn't quite understand what software was, and thought it had something to do with mathematical equations.[13] Mathematical equations and laws of nature are not patentable, so the court held software was not patentable.

Eventually the Supreme Court and lower courts worked out a compromise position where they concluded that once they understood what software really is, it will be covered by the patent laws, but only if it's tied to something physical, tied to a machine.[14] The holding was that you can't simply patent an algorithm or computer program in the abstract, it had to be connected very intimately to some hardware. That doctrine has begun to break down in the past three to four years. For example, we're now at the point where the patent office, it appears, is willing to accept a computer program on a floppy disk as an article of manufacture and say that's patentable.[15] Where is the hardware it's supposed to be tied to? It's on the disk. The disk is the hardware, and the disk has some magnetic flux on it that encodes the program and that should be patented.Data structures, ways of arranging the information within a computer memory also appear to be patentable. And so that requirement that the Supreme Court said we had to have to tie software to hardware appears to be breaking down.

---

[11]*See, e.g.*, Computer Assocs. Int'l v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992).

[12]35 U.S.C. § 101.

[13]*See* Parker v. Flook, 437 U.S. 584 (1978); Gottschalk v. Benson, 409 U.S. 63 (1972).

[14]*See* Diamond v. Dieher, 450 U.S. 175 (1981); *In re* Alappat, 33 F.3d 1526 (Fed. Cir. 1994).

[15]*See In re* Beauregard, 53 F.3d 1583 (Fed. Cir. 1995) (dismissing appeal from PTO rejection of software on a floppy disk).

What does that mean? That means a couple of traditional patent doctrines that would have prevented what I think is going to be a train wreck with the First Amendment are no longer available. One was the so-called mental steps doctrine which said that as you were going to claim your invention in patent law, you could not have as part of your claim mental processes.[16] Suppose my invention is that you move this lever and switch this switch and then you sort of add some numbers in your head and move it again. Traditionally that wasn't permissible, in part because it was too indefinite. We didn't know what the invention was or what was going on in the head of the person. But, in addition, people said there's some sort of concern here about maybe people thinking patented thoughts. If part of the process you're claiming is something going on in your head, first of all, how are we ever going to enforce that patent? Because we don't know what's going on in your head, and policing that claim would raise serious privacy concerns. In addition, we have questions about the constitutionality of that claim. Can you prevent mental processes? Can you give people the right to exclude folks from thinking certain types of thought or going through mental steps in their head? But the mental steps doctrine has slowly been whittled away as courts saw things could be implemented in machines that would otherwise go on in people's head, so we don't talk about mental steps much anymore.

Nor do we talk about the closely related printed matter doctrine, which said that you cannot claim text as part of your invention.[17] So if my invention is a machine that has words printed on the side of it, traditionally the law would have said you can't claim the writing on the machine as your invention. This typically came up in the case of business forms. But as I showed a moment ago, now we're at the point where we can submit a floppy disk to the patent office and they think that's okay. The patent office objected a couple of years ago under the printed matter doctrine. You sent us a floppy disk that had magnetic flux on the disk, it's just as if you have writing on your invention and we're not going to accept that under patent law. But as I said, the patent office seems to have acquiesced to that kind of claim, so printed matter is no longer a barrier.

What does that mean to us in terms of *Bernstein*? Well, if software is really speech and we know that software really is patentable in most situations, we can't rely on the mental steps doctrine these days or printed matter doctrine to

[16]*See generally* 1 D. CHISUM, CHISUM ON PATENTS, § 1.03[6] (1997) (collecting cases); *see also generally* Pamela Samuelson, *Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 EMORY L.J. 1025 (1990) (reviewing and critiquing mental steps cases).

[17]*See generally* 1 D. CHISUM, *supra* note 16 at 1.02[4] (collecting printed matter cases).

prevent what appears to be a conflict with the First Amendment. We're going to give people exclusive rights in something that appears, according to the *Bernstein* court, to be speech. Remember that patent law, unlike copyright, is not designed to accommodate First Amendment interests. We've tended to say there's not much of a conflict between copyright and the First Amendment because we have the "fair use" sort of safety valve, that doctrine that allows people to express things.[18] But we all assumed that patent law was about machines and tangible things, so no fair use doctrine ever developed.

So where does this leave us then in dealing with the applications of *Bernstein*? It doesn't look like source code/object code distinction will allow us to separate speech from function, as we've seen from experience in the copyright law, the distinction doesn't make a whole lot of sense for software. If we take the *Bernstein* analogy to copyright seriously, that distinction is going to break down very quickly, leading to patenting of speech. Can we amend the patent law perhaps that we have some sort of fair use in patents for software? If software is expression, it really is speech, the First Amendment may force us to do that. Alternatively, we may need to come up with some kind of hybrid model to classify software. There has long been agitation for sui generis protection, or brand new type of protection besides copyright, patent, or trade secrecy, that accommodates software, since it is a strange sort of creature. Perhaps we need to have some new model, not just for intellectual property in software, but for First Amendment interests in software, to deal with software if it really is speech. Otherwise I'm afraid we're going to see if the First Amendment jurisprudence will have the same type of distortions and difficulty that we see in copyright from trying to force software into the copyright expression box.

[18]*See, e.g.*, Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569 (1994); Harper & Row Publishers, Inc. v. Nation Enters., 471 U.S. 539 (1985).