FIGHTING KEYWORDS: TRANSLATING THE FIRST AMENDMENT TO PROTECT SOFTWARE SPEECH

Robert Plotkin[*]

The ongoing debate over the applicability of the First Amendment to software focuses primarily on whether software is speech, a device, or a combination of both. According to the terms of this debate, if software is speech then the First Amendment fully protects it; however, if software is a device, it deserves no First Amendment protection. I argue that this debate is not comprehensive because the mere classification of software as "speech" or as a "device" does not end the First Amendment inquiry. I propose an alternative framework in which well-accepted principles of tort law, criminal law, and First Amendment jurisprudence are combined to provide maximum protection for "software speech," while contemporaneously promoting the public interest in regulating harm. Shaping the precise contours of such a framework, however, will require the resolution of difficult public policy questions raised by the unique nature of software and the Internet.

I. INTRODUCTION

A. Software Runs Into the Law

Computer programs[1] typically are designed and implemented using source code[2] written in a computer programming language.[3] The source

[*] Robert Plotkin, Esq. is an intellectual property attorney with a solo practice based in Concord, Massachusetts. He is also a Lecturer at Boston University School of Law, where he teaches a seminar entitled "Software and the Law." He is licensed to practice law in Massachusetts and New York, and is registered to practice before the United States Patent and Trademark Office. Attorney Plotkin received his bachelor's degree in Computer Science and Engineering from Massachusetts Institute of Technology and his law degree from Boston University School of Law. The author wishes to thank Melissa Hoffer for the insightful and invaluable feedback she provided on the topics addressed by this Article. The author also wishes to thank Meleena Bowers and Cynthia Gilbert for research assistance provided in the preparation of this Article.

code for a particular program specifies instructions that may be provided to a computer to be executed.[4]

The same source code that is provided to a computer to create an executable computer program may also be read and understood by computer programmers.[5] In fact, computer programmers often use source code to convey ideas about program design and to communicate generally with each other about their programs.[6] Therefore, the source

[1] The terms "software" and "computer programs" are used interchangeably herein. For dictionary definitions of "software," see MERRIAM WEBSTER'S COLLEGIATE DICTIONARY 1117 (10th ed. 1993) (defining "software" as "something used or associated with and usu[ally] contrasted with hardware," such as "the entire set of programs, procedures, and related documentation associated with a system and esp[ecially] a computer system"); MICROSOFT COMPUTER DICTIONARY 489 (5th ed. 2002) (defining "software" as "[c]omputer programs; instructions that make hardware work").

[2] "Source code" may be defined as "[h]uman-readable program statements written by a programmer or developer in a high-level or assembly language that are not directly readable by a computer." MICROSOFT COMPUTER DICTIONARY, *supra* note 1, at 491.

[3] A "programming language" is "[a]ny artificial language that can be used to define a sequence of instructions that can ultimately be processed and executed by [a] computer." MICROSOFT COMPUTER DICTIONARY, *supra* note 1, at 426. Well-known examples of programming languages include BASIC, C, C++, Pascal, and Java.

[4] In the context of computer programs, the term "execute" means "[t]o perform an instruction." MICROSOFT COMPUTER DICTIONARY, *supra* note 1, at 200.

[5] This observation follows from the definition of source code. *See supra* note 2. *See also* Universal City Studios, Inc. v. Corley, 273 F.3d 429, 446 (2d Cir. 2001) ("[t]he undisputed evidence reveals that even pure object code can be, and often is, read and understood by experienced programmers. And source code . . . can be read by many more.") [hereinafter *Corley*]; Junger v. Daley, 209 F.3d 481, 484 (6th Cir. 2000) ("[F]or individuals fluent in a computer programming language, source code is the most efficient and precise means by which to communicate ideas about cryptography."); Bernstein v. United States Dep't of Justice, 176 F.3d 1132, 1142 (9th Cir. 1999) [hereinafter *Bernstein IV*] ("source code is not meant solely for the computer, but is rather written in a language intended also for human analysis and understanding"); United States v. Elcom Ltd., 203 F. Supp. 2d 1111, 1126 (N.D. Cal. 2002) (holding that software is speech protected by the First Amendment in both the object code and the source code formats); Brief of Amici Curiae Dr. Harold Abelson et al. 6, *Corley* (No. 00-9185) *available at* http://www.eff.org/Cases/MPAA_DVD_cases/20010126_ny_progacad_amicus.html (last visited Apr. 15, 2004) ("Computer codes are text languages, as expressive as any text language, with dialects, grammar structures and nuances, and thus are entitled to the same level of First Amendment scrutiny as any natural text language, such as English.").

[6] *See Junger*, 209 F.3d at 484; *Elcom*, 203 F. Supp. 2d at 1126; *Corley*, 273 F.3d at 448 ("Programmers communicating ideas to one another almost inevitably communicate in code, much as musicians use notes."); Brief of Amici Curiae Dr. Harold Abelson et al., *Corley* (No. 00-9185), *supra* note 5 ("Among academics and programmers, communicating in computer code . . . is essential '[t]o promote the Progress of Science and useful Arts . . .', the core purpose of copyright.");

> Writing, analyzing and publishing cryptographic algorithms and software is integral to [Professor Bernstein's] academic research and teaching. . . . Software in the form of source code was designed to be read and understood by humans and is a critical tool in teaching on subjects involving computers. It is as difficult to develop the science of cryptography without reading software as it would be to develop poetry without reading poems or the theory of relativity without reading mathematical equations.

Brief for the Appellee at 2, *Bernstein v. U.S. Dep't of State*, 922 F. Supp. 1426 (N.D. Cal. 1996) (No. 97-16686) *available at* http://www.eff.org/Privacy/Crypto_export/Bernstein_case/Legal/971110_bernstein.appeal; Steven E. Halpern, *Harmonizing the Convergence of Medium, Expression, and Functionality: A Study of the Speech Interest in Computer Software,* 14 HARV. J. L. & TECH. 139, 142–44 (2000) (explaining that source code is intended to, and actually does both: (1) communicate information to those who read the source code; and (2) operate on a computer to create a result); Lee Tien, *Publishing Software as a Speech Act*, 15 BERKELEY TECH. L.J. 629, 629 (2000) ("publishing code for a particular program may serve both as a vehicle for a computer to carry out its instructions and for the expression of information to human programmers.[7]

The First Amendment traditionally has been applied to conventional forms of oral and written expression such as speeches, novels, poems, and scientific research papers. Computer source code is similar to these forms of expression in that source code may be used by programmers to communicate ideas with one another. In particular, the act of distributing source code is similar to the act of speaking in a natural language to the extent that both acts intentionally convey information.[8] Therefore, it should not be surprising that the courts have been called on to address the nature and extent of the First Amendment's applicability to source code in cases in which the actual or potential distribution of source code has formed the factual basis for a legal claim.[9]

In recent years, for example, computer programmers have brought First Amendment challenges to several laws that are capable of being applied prospectively to limit the distribution of software and/or to make distributors of software (e.g., programmers) liable for damages. The programmers have argued that the laws in question implicate the First Amendment because software, particularly in source code form, is a kind of speech that programmers use to communicate with each other for scientific, academic, and other professional purposes.

For example, federal regulations limiting the export of encryption technology have been challenged on the ground that such regulations violate the First Amendment rights of cryptographic professionals,[10] in

source code generally is a speech act because computer scientists and programmers conventionally intend to communicate ideas about computational procedures by publishing source code").

[7] *See Corley*, 273 F.3d at 448 ("Instructions such as computer code, which are intended to be executable by a computer, will often convey information capable of comprehension and assessment by a human being.").

[8] *See* Tien, *supra* note 6, at 671 ("Programming languages make it possible to express exceedingly precise, particularized meanings. Computer programs are statements in languages peculiarly suited for expressing procedures and ideas about procedures. For this reason, asserting source code is a speech act.").

[9] *See infra* notes 10, 12.

[10] The statutory and regulatory framework governing export of encryption technology is convoluted and has changed during the course of the relevant cases. *See* Bernstein v. U.S. Dep't of State, 945 F. Supp. 1279, 1283–84 (N.D. Cal. 1996) [hereinafter *Bernstein II*]; Bernstein v. U.S. Dep't of State, 974 F. Supp. 1288, 1291–92 (N.D. Cal. 1997) [hereinafter *Bernstein III*]. For purposes of simplicity, the various statutes and implementing regulations that have been the subject of litigation in this area are referred to herein as the "encryption export regulations."

In Bernstein v. U.S. Dep't of State, 922 F. Supp. 1426, 1439 (N.D. Cal. 1996) [hereinafter *Bernstein I*], a mathematician challenged the enforcement of the International Traffic in Arms Regulations ("ITAR") as applied to his encryption software source code in both print and electronic form. The plaintiff alleged that the ITAR and the statute under which they were enacted were, both facially and as applied: (1) content-based infringements on speech; (2) unconstitutional prior restraints on speech; (3) vague; (4) overbroad; and (5) in violation of the rights of association and equal protection. The *Bernstein I* court held that both computer source code in general and the cryptographic source code specifically in question are protected speech under the First Amendment, and that the ITAR constituted an unconstitutional prior restraint on expression. In *Bernstein II*, the particular, their right to publish the source code to encryption software as part of professional and academic discourse on cryptographic techniques. More recently, the Digital Millennium Copyright Act ("DMCA") has been challenged on the ground that its prohibition against trafficking in devices that circumvent certain copy-protection and access-control measures[11] violates the First Amendment insofar as it prohibits the distribution of software code for performing such functions.[12] The U.S. Supreme Court has not yet addressed the

court granted summary judgment for Bernstein on his First Amendment claims. The Department of Commerce took control over licensing authority for nonmilitary encryption technologies and Bernstein amended his complaint to raise the same challenges to the Export Administration Regulations ("EAR") in *Bernstein III*. The Court granted summary judgment for Bernstein, enjoining the Commerce Department from enforcing the invalidated provisions. The Government appealed in *Bernstein IV*, where the court again held for Bernstein. *See generally Bernstein IV*, *supra* note 5. In September 1999, the Ninth Circuit held in Bernstein v. U.S. Dep't of Justice, 192 F.3d 1308 (9th Cir. 1999) [hereinafter *Bernstein V*], that the case would be reheard by the court *en banc*. That rehearing has not yet taken place.

In *Junger*, a Case Western University School of Law professor brought suit in the Eastern District of Ohio challenging the constitutionality of the EAR as applied to the encryption software that he sought to distribute via a Web site. The district court held that source code is conduct because it is purely functional in most circumstances, while recognizing that source code may be expressive in limited circumstances. The court held that the export of encryption source code is therefore not protected by the First Amendment, and that the encryption export regulations therefore do not constitute a prior restraint on expression. The Sixth Circuit reversed and remanded for further proceedings, finding that computer source code is protected by the First Amendment because it "is an expressive means for the exchange of information and ideas about computer programming." *Junger*, 209 F.3d at 484–85.

In *Karn v. U.S. Dep't of State*, an exporter challenged the application of the encryption export regulations to a computer diskette containing cryptographic software. The *Karn* court held that the software at issue was speech but determined that the regulations were content-neutral and satisfied the "intermediate scrutiny" that is applied to content-neutral regulations. 925 F. Supp. 1, 11 (D.D.C. 1996).

For more information about the encryption cases, see generally, David McClure, Note, *First Amendment Freedoms and the Encryption Export Battle: Deciphering the Importance of* Bernstein v. United States Department of Justice, 79 NEB. L. REV. 465 (2000); Katherine A. Moerke, *Free Speech to a Machine? Encryption Software Source Code is Not Constitutionally Protected "Speech" Under the First Amendment*, 84 MINN. L. REV. 1007 (2000); Norman Andrew Crain, Bernstein*,* Karn*, and* Junger*: Constitutional Challenges to Cryptographic Regulations*, 50 ALA. L. REV. 869 (1999).

[11] The DMCA prohibits, *inter alia*, trafficking in devices that circumvent technologies for controlling access to or preventing copying of copyrighted works. It states that:

> (a) . . .
> (1)(A) No person shall circumvent a technological measure that effectively controls access to a work protected under this title; . . .
> (2) No person shall manufacture, import, offer to the public, provide, or otherwise traffic in any technology, product, service, device, component, or part thereof, that -
> . . . is primarily designed or produced for the purpose of circumventing a technological measure that effectively controls access to a work protected under this title . . .
> (b) . . . (1) No person shall manufacture, import, offer to the public, provide, or otherwise traffic in any technology, product, service, device, component, or part thereof, that—
> (A) is primarily designed or produced for the purpose of circumventing protection afforded by a technological measure that effectively protects a right of a copyright owner under this title in a work or a portion thereof.

17 U.S.C. § 1201 (2000).

[12] In *United States v. Elcom Ltd.*, the defendant moved to dismiss charges alleging violations of the DMCA's criminal anti-circumvention provisions. The charges were brought against the defendant based on its sale of a software program that enables users to access, distribute, copy, and print electronic books in which access and copy controls have been implemented. United States v. Elcom, Ltd., 203 F. Supp. 2d 1111, 1122 (N.D. Cal. 2002). In *Corley*, the defendants argued that they were

applicability of the First Amendment to source code, and the lower courts have yet to develop a comprehensive or even internally consistent set of rules to apply in such cases.

B. Two Camps in Caricature

Although litigants and commentators have put forth a wide variety of arguments both for and against the applicability of the First Amendment to source code, the proponents of the arguments on each side may generally be divided into two camps: the "software-as-speech" camp and the "software-as-device" camp. I will describe the positions of each camp in gross caricature to present such positions in their purest form, and therefore simplify and clarify the discussion that follows, recognizing that such caricatures gloss over distinctions within and between the camps.

Each of the two camps: (1) asserts an ontological position on what the nature of software (particularly in source code form) is; and (2) infers, from the asserted nature of software, a legal conclusion about whether and to what extent the First Amendment should protect expression embodied in software. In particular, the software-as-speech camp takes the position that computer programs, whether embodied in source code or in some other form, are "pure speech" used by computer programmers to express ideas to each other.[13] This camp essentially

engaged in protected speech when they posted software code, and links to such code, for decrypting and copying encrypted DVDs. 273 F.3d at 436. In *DVD Copy Control Ass'n v. Bunner*, plaintiffs who licensed decryption technology to manufacturers of hardware and software for playing DVDs sought an injunction against a republisher of computer code for circumventing the plaintiffs' decryption technologies. DVD Copy Control Ass'n v. Bunner, 113 Cal. Rptr. 2d. 338, 340 (Cal. Ct. App. 2001), *rev'd* 31 Cal. 4th 864 (2003). On remand from the California Supreme Court, the California Court of Appeals held that the record evidence failed to support a preliminary injunction for trade secret misappropriation, and therefore dissolved the injunction on the grounds that it burdened more speech than necessary to protect the plaintiffs' property interest and was an unlawful prior restraint upon the defendant's right to free speech. *Bunner,* 113 Cal. Rptr. 2d. at 349. *See also* Plaintiff's Complaint at para. 52, Edelman v. N2H2, Inc., 263 F. Supp. 2d 137 (D. Mass. 2003) (No. 02-11503), *available at* http://archive.aclu.org/court/edelman.pdf (last visited Apr. 15, 2004) ("The license and statutory provisions are chilling core scientific speech by Mr. Edelman on a matter of public interest to the detriment of his First Amendment rights.").

[13] The court in *Corley* stated:

> The Appellants vigorously reject the idea that computer code can be regulated according to any different standard than that applicable to pure speech, i.e., speech that lacks a nonspeech component. Although recognizing that code is a series of instructions to a computer, they argue that code is no different, for First Amendment purposes, than blueprints that instruct an engineer or recipes that instruct a cook.

Corley, 273 F.3d at 451; *Bunner*, 113 Cal. Rptr. 2d at 349 ("Although the social value of DeCSS may be questionable, it is nonetheless pure speech."). The Appellee's Brief in *Bernstein IV* stated:

> [T]he author of the seminal work on computer programming, Professor Donald E. Knuth of Stanford University, wrote: "Programming is best regarded as the process of creating works of literature, which are meant to be read . . . Computer programs that are truly beautiful, useful, and profitable must be readable by people. So we ought to address them to people, not to machines."

Brief for Appellee, at 33, *Bernstein IV* (No. 97-16686), *available at* http://cr.yp.to/export/1997/1110-cohn.txt (last visited Apr. 15, 2004)*; see also* Brief of Amici Curiae Dr. Harold Abelson et al., at Part II, *Corley* (No. 00-9185), *supra* note 5 ("At root, computer code is nothing more than text, which, like denies that computer source code is "functional" or a "device."[14] The programmers in the cryptography and DMCA cases cited above are good representatives of this camp.[15]

Put simply, the software-as-speech camp asserts that software should be fully protected by the First Amendment because software is pure speech.[16] Note that this theory includes both an assertion that software should be protected by the First Amendment and that software should receive the highest available degree of First Amendment protection. To the extent that the software-as-speech camp admits that software may be functional in some respects, and therefore differs from

any other text, is a form of speech."); Brief of Amicus Curiae American Association for the Advancement of Science, at Part I.B, *Bernstein IV* (No. 97-16686), *available at* http://cr.yp.to/export/1997/1110-marks.txt (last visited Apr. 15, 2004) ("Source code is not conduct, however, but speech in its purest form. It is people talking to one another using a complex system of mutually understood meanings."); EFF Supplemental Letter Brief, at Part A.3, *Corley* (No. 00-9185), *available at* http://www.eff.org/IP/Video/MPAA_DVD_cases/20010530_ny_eff_supl_brief.html (last visited Apr. 15, 2004) ("The dissemination of DeCSS, here by a member of the media covering an issue of public concern, is pure speech.").

[14] *See, e.g., Corley*, 273 F.3d at 457 ("As they have throughout their arguments, the Appellants ignore the reality of the functional capacity of decryption computer code and hyperlinks to facilitate instantaneous unauthorized access to copyrighted materials by anyone anywhere in the world."); EFF Supplemental Letter Brief, at Part A.2, *Corley* (No. 00-9185), *supra* note 13 ("DeCSS itself has no non-speech elements. It is a set of instructions written in a specific professional language that expresses ideas to those who can read that language. Computer programmers and scientists communicate using programming languages because these languages are an unambiguous mode of expression.").

> The court below properly recognized that computer code, whether in the form of source code or object code, is a form of expression, subject to First Amendment scrutiny. However, based in part on the court's ill-considered concerns about the 'functionality' of code, . . . the court ruled that code is entitled only to the intermediate level of scrutiny set forth in *United States v. O'Brien*, 391 U.S. 367 (1968), rather than to the strict judicial scrutiny which should be afforded to this constitutionally protected speech.

Brief of Amici Curiae Dr. Harold Abelson et al., at 6, *Corley* (No. 00-9185), *supra* note 5; Brief for Amicus Curiae Garrett Epps, at Part II.D, *Bernstein I* (No. 97-16686), *available at* http://cr.yp.to/export/1997/1110-epps.txt (last visited Apr. 15, 2004) ("Like a book, a floppy disk may encourage further conduct but alone it is incapable of action. Its functionality is that of a piece of paper and a pen; it is a surface on which ideas and expression can be captured and shared with others.").

[15] *See supra,* notes 10–12.

[16] In *Karn*, for example, the plaintiff argued (unsuccessfully) that "the O'Brien criteria are inapplicable because they apply only to the regulation of 'conduct,' and that the Karn diskette [containing software] is 'pure speech,' the regulation of which should require strict scrutiny review." Karn v. U.S. Dep't of State, 925 F. Supp. 1, 10 (D.D.C. 1996). The *Corley* court also stated:

> The Appellants vigorously reject the idea that computer code can be regulated according to any different standard than that applicable to pure speech, i.e., speech that lacks a nonspeech component. Although recognizing that code is a series of instructions to a computer, they argue that code is no different, for First Amendment purposes, than blueprints that instruct an engineer or recipes that instruct a cook.

Corley, 273 F.3d at 451.

At least one court has adopted the position that source code, as a category, should not be analyzed differently under the First Amendment than other kinds of speech. The *Bernstein I* court stated, "[n]or does the particular language one chooses change the nature of language for First Amendment purposes. This court can find no meaningful difference between computer language, particularly high-level languages as defined above, and German or French. All participate in a complex system of understood meanings within specific communities." *Bernstein I*, 922 F. Supp. 1426, 1439 (N.D. Cal. 1996).

other kinds of speech, the functional aspects of software are asserted to be *de minimis*.[17]

The software-as-device camp, in contrast, takes the position that the source code for a computer program is functional (i.e., it performs a useful function) in the same manner as more conventional mechanical and electrical devices.[18] For example, this camp considers software that cracks an encryption scheme to be identical to a physical key or a set of burglar's tools.[19] According to this camp, software is designed and distributed to perform functions like any other machine, not to express ideas.[20]

[17] The EFF Supplemental Letter Brief for Corley stated:

> That a person might use a computer program to do something does not by itself add "nonspeech" elements to the text. Blueprints and instructions for a photocopier, recipes, books about fixing cars, and videos on baby care surely count as fully protected speech, no matter how "functional" their content might prove to be. Prohibition of these works merely because recipients might easily do what they describe or explain would require searching First Amendment review and justification.

EFF Supplemental Letter Brief, at Part A.2, *Corley* (No. 00-9185), *available at* http://www.eff.org/IP/Video/MPAA_DVD_cases/20010530_ny_eff_supl_brief.html (last visited Apr. 15, 2004). The brief for Defendant-Appellant in *Corley* stated:

> Text expressed in a computer programming language is an important medium of communication of scientific theories and ideas among scientists and is therefore protected under the First Amendment. The government's *post hoc* characterization of cryptographic source code as a "product" as opposed to "information" does not insulate such text from First Amendment protection.

Brief for Amici Curiae EPIC et al., at Summary of Argument, *Bernstein I* (No. 97-16686), *available at* http://www.epic.org/crypto/export_controls/bernstein_brief.html (last visited Apr. 15, 2004).

[18] *See, e.g.,* Brief for Plaintiffs-Appellees, at Part B.1, *Corley* (No. 00-9185), *available at* http://www.eff.org/IP/Video/MPAA_DVD_cases/20010228_ny_op_reply_brief.html (last visited Apr. 15, 2004) (arguing that DeCSS is simply a tool for decrypting DVDs, having no expressive content itself, and should therefore be subject to the same analysis as would be accorded a key, a password, or a virtual machine); United States v. Elcom Ltd., 203 F. Supp. 2d 1111, 1126 (N.D. Cal. 2002) (order denying defendant's motions to dismiss the indictment on constitutional grounds, summarizing the government's arguments that the DMCA targets the sale of technology, not speech, and that the software developed and sold by the defendant is not speech protected by the First Amendment).

[19] The *Corley* court stated:

> In its basic function, CSS is like a lock on a homeowner's door, a combination of a safe, or a security device attached to a store's products. DeCSS is computer code that can decrypt CSS. In its basic function, it is like a skeleton key that can open a locked door, a combination that can open a safe, or a device that can neutralize the security device attached to a store's products.

Corley, 273 F.3d at 452–53; MPAA Supplemental Letter Brief, at IV.2, *Corley* (No. 00-9185), *available at* http://www.eff.org/IP/Video/MPAA_DVD_cases/20010530_ny_mpaa_supl_brief.html (last visited Apr. 15, 2004) ("DeCSS is not information within the meaning of the First Amendment, much less a message; it is, in the government's apt phrase, a 'digital crowbar,' a technical device for circumventing protection that copyright owners have installed on copies of their copyrighted works.").

[20] *See Karn*, 925 F. Supp. at 9 n.19 ("Source codes are merely a means of commanding a computer to perform a function."); *Bernstein I*, 922 F. Supp. at 1434 ("According to defendants, the source code, as a functioning cryptographic product, is not intended to convey a particular message. It cannot be speech, they say, because its purpose is functional rather than communicative."). The MPAA Supplemental Letter Brief for Corley stated:

> DeCSS is a device . . . that accomplishes a mechanical task, namely descrambling and decrypting an encrypted, scrambled DVD and copying its content to a hard drive. It is no more "speech" than a key to a library or museum (or a crowbar that could force open their doors) is "speech."

MPAA Supplemental Letter Brief, at Part IV.2, *Corley* (No. 00-9185), *supra* note 19. *See also* Brief for Plaintiffs-Appellees, at Part I.B, *Corley* (No. 00-9185), *supra* note 18 ("The DMCA bars trafficking in circumvention devices (regardless of how configured) because of what they do and are suited to do, not because of their ideas; indeed, like the 'black boxes,' 'passwords,' and keys which Congress

The software-as-device camp asserts that software should not receive any First Amendment protection because software is not expressive.[21] In particular, it is argued that statutes regulating the distribution of devices that perform specified functions should be subject, at most, to intermediate scrutiny under the *O'Brien*[22] test when such regulations are applied to software.[23] To the extent that the software-as-device camp admits that software may be expressive in some respect, and therefore different from other kinds of devices, it asserts that the expressive aspects of software are either *de minimis* or do not qualify as the kind of expression that is protected by the First Amendment.[24]

C. Problems with Conventional Positions

The arguments made by the two camps described above have at least two flaws in common.

1. Adoption of the Function-Expression Dichotomy

Both the software-as-speech and software-as-device camps at least implicitly adopt the function-expression dichotomy, according to which a particular work may either be functional or expressive, but not both.[25] A

envisioned, decryption devices express no 'ideas.'"); Moerke, *supra* note 10, at 1009 (arguing that encryption software source code is not a form of expression); R. Polk Wagner, Note, *The Medium is the Mistake: The Law of Software for the First Amendment*, 51 STAN. L. REV. 387, 407 (1999) ("The analysis of software under the speech-conduct distinction is problematic because computer code is primarily, perhaps even exclusively, functional. . . . [S]oftware, to the extent it can be considered a set of instructions, communicates only to machines.") .

[21] *See* Brief for Plaintiffs-Appellees, at Part I.B, *Corley* (No. 00-9185), *supra* note 18 ("Decryption devices express no 'ideas.'"); Universal City Studios, Inc. v. Reimerdes, 82 F. Supp. 2d 211, 219 (S.D.N.Y. 2000) ("As a preliminary matter, it is far from clear that DeCSS is speech protected by the First Amendment. In material respects, it is merely a set of instructions that controls computers."). In their brief, Plaintiffs-Appellees in *Corley* argued that:

> Corley's unauthorized provision to the public of burglary tools, a password, or an automobile key would not implicate the First Amendment, and no different analysis should apply merely because the decrypting technology Corley provided happens to have been configured as software; indeed, Congress prohibited trafficking in decryption devices whether configured as hardware or software and, accordingly, this Court should reach the same result as the Ninth Circuit did in *Mendelsohn* [by declining to apply any First Amendment review].

Brief for Plaintiffs-Appellees, at Part I, *Corley* (No. 00-9185), *supra* note 18.

[22] United States v. O'Brien, 391 U.S. 367 (1968).

[23] Brief for Plaintiffs-Appellees, at Part I, *Corley* (No. 00-9185), *supra* note 18 (arguing that the DMCA should be subject to intermediate scrutiny, at most, when applied to the distribution of DeCSS).

[24] *See, e.g., Bernstein I*, 922 F. Supp. at 1432–33 ("Defendants maintain that plaintiff has raised no colorable constitutional claim because this case does not concern 'speech' protected by the First Amendment, and, even if it does, the minimal infringement is excusable under *O'Brien v. United States*"); *see generally* MPAA Supplemental Letter Brief, *Corley* (No. 00-9185), *supra* note 19 (arguing that the source code at issue does not qualify as political speech or any other kind of protected speech).

[25] *See, e.g.,* David S. Touretzky, *Free Speech Rights for Programmers*, COMM. OF THE ACM, Aug. 2001, at 23–24 (arguing that, in the eyes of programmers, code is speech rather than a device, implicitly assuming that code may not be both speech and a device).

work is "expressive" if it conveys an idea. Novels, poems, dances, and political speeches are examples of expressive works because they communicate or express ideas, whether such ideas be facts, emotions, opinions, or otherwise. A work is "functional" if it performs a useful purpose in the physical world.[26] Physical tools and machines, such as hammers, automobiles, and light bulbs are quintessential examples of "functional" objects. A "function" in this sense typically refers to the manipulation of matter and/or energy to achieve a useful result. According to this definition of "functional," purely aesthetic functions (such as the "function" of expressing an idea) are not considered to be functions. A poem, for example, is not "functional" even though it performs the putative "function" of expressing an idea.[27]

At least prior to the advent of software, the function-expression dichotomy was often fairly useful as a heuristic device, because it was often the case that a particular work either was expressive or functional, but not both.[28] A hammer, for example, performs the function of pounding nails and does not express an idea, at least not through its intended and typical use. Conversely, a poem expresses an idea but does not directly perform a function, as the term is defined above. As such, the function-expression dichotomy accurately reflects the properties of many classes of works, and therefore is often useful as a heuristic device. For instance, upon ascertaining that a particular work (such as a poem) is expressive, one may accurately conclude according to the function-expression dichotomy that the work is not functional, and vice versa.

There are, however, many exceptions to this rule; a single work may express an idea as well as perform a function. A work of architecture, such as an arch, may be both expressive and functional.[29] Similarly, a single political action, such as a picket line, may express the picketers'

The function-expression dichotomy also arises in the context of intellectual property law. For example, Professor Pamela Samuelson has argued that:

> [p]rograms are . . . too much of a mechanical process to fit comfortably in the copyright system and too much of a writing to fit comfortably in the patent system. They are a hybrid – both writing and machine at the same time – in a legal system that has generally assumed that an intellectual product is either a writing (and hence copyrightable) or a machine (and hence patentable), but not both at the same time.

Pamela Samuelson, Benson *Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 EMORY L.J. 1025, 1128–29 (1990).

[26] For a more detailed explanation of how the term "functional" is used herein, see *infra* Part IV.E.

[27] This distinction between functional (utilitarian) works and expressive works draws heavily from intellectual property law, in which patent law protects functional works while copyright law protects expressive works. *See generally* Robert Plotkin, *Computer Programming and the Automation of Invention: A Case for Software Patent Reform,* 2003 UCLA J. L. TECH. 7 (2003).

[28] *But see* Tien, *supra* note 6, at 629 (arguing that instead of asking whether something such as software is speech, we should ask whether someone is speaking, using "speech act" theory as our guide).

[29] *Id.* at 644 ("The artist Marcel Duchamp's famous readymade sculpture, *The Fountain*, which is simply a urinal, is speech because the shared conventions of the art world, as expressed by the medium of an art exhibit, made it so.").

opposition to a company policy while performing the physical function of blocking the entrance to a factory.

The two camps in the software debate described above seem to accept the function-expression dichotomy quite strictly. The software-as-speech camp identifies expressive aspects of software and concludes that, because software is expressive, it cannot also be functional. Similarly, the software-as-device camp identifies functional aspects of software and concludes that, because software is functional, it cannot be expressive. A third possibility exists, however. Software may be both expressive and functional, perhaps in degrees that vary from case to case. At least some of the parties to this debate are aware of this third possibility, but nonetheless explicitly reject it.[30]

The source code for a particular program may have both a functional aspect and an expressive aspect.[31] The source code may be functional in the sense that it is capable of causing a machine—a general-purpose computer programmed with the source code—to perform a useful function. The source code is functional in the same way as a drill bit, which, when coupled to a drill, is capable of causing the drill to perform a particular function as defined by the shape of the bit.[32]

The source code may be expressive, just as a novel, poem, or architectural blueprint, in the sense that it is capable of expressing ideas to human programmers. It is possible for the programmer to use the source code to convey ideas to other programmers even if the code is never used to create an executable software program.

A particular piece of source code may also be both *highly* expressive and *highly* functional.[33] Source code may be highly expressive in the sense that it expresses a wide variety of complex ideas unambiguously, and software may be highly functional by reliably and repeatedly performing a wide variety of complex functions.

Thus, software is a new beast. Historically, there have been many examples of objects and actions which are both expressive and

[30] *See supra* Part I.B and sources cited therein.

[31] *See, e.g.,* Junger v. Daley, 209 F.3d 481, 484 (6th Cir. 2000) ("The issue of whether or not the First Amendment protects encryption source code is a difficult one because source code has both an expressive feature and a functional feature."). Some have argued that the dual functional-expressive nature of software implies that software should be protected by a hybrid or *sui generis* system of intellectual property protection. *See generally* Ejan Mackaay, *Legal Hybrids: Beyond Property and Monopoly?*, 94 COLUM. L. REV. 2630 (1994); J.H. Reichman, *Legal Hybrids Between the Patent and Copyright Paradigms*, 94 COLUM. L. REV. 2432 (1994); John Swinson, *Copyright or Patent or Both: An Algorithmic Approach to Computer Software Protection*, 5 HARV. J.L. & TECH. 145 (1991); Daniel G. Feder, Comment, *Computer Software: Hybrid Technology Demanding a Unique Combination of Copyright and Patent Protection*, 59 U. MO. KAN. CITY L. REV. 1037 (1991); John M. Griem, Jr., Note, *Against a* Sui Generis *System of Intellectual Property for Computer Software*, 22 HOFSTRA L. REV. 145 (1993); Gregory J. Maier, *Software Protection – Integrated Patent, Copyright and Trade Secret Law*, 28 IDEA 13 (1987).

[32] *See generally* Plotkin, *supra* note 27.

[33] *See* Wagner, *supra* note 20, at 407 ("Indeed, if there ever was conduct that was an 'undifferentiated whole' of action and expression, the use and operation of computer code would seem to be it.").

functional, but typically it has not been possible to embody a high degree of expressiveness and functionality in the same object or action. Although engineers have long attempted to express ideas through machine designs, complex (highly functional) machines have had a minimal expressive aspect. Similarly, highly expressive artwork has been limited in functionality.

In particular, to express ideas that are both complex and precise, it has been necessary to resort to writing, or some other form of linguistic expression, and to abandon hope of embodying the message in a machine or other functional device. Although a great work of art may express a powerful message, such messages are somewhat ambiguous and subject to interpretation. Works of art are not capable of conveying ideas with the same precision as mathematical notation or other formal symbolic languages.

With software, however, it is possible to express ideas with the full complexity and precision of mathematics and to perform an extremely wide variety of extraordinarily powerful functions. For the first time in history, an entire category of human artifact is characterized by the capability to embody both features. The complexity and precision of expression and function that may be embodied in a single software program is limited only by the power of the programming languages and the computers we develop. It is likely that the expressiveness and functionality of software will continue to grow over time.

2. Leaps of Logic

A second problem with the positions taken by the two opposing camps is that each jumps from its ontological premise (software is expressive or software is functional) to a particular legal conclusion (software should be fully protected by the First Amendment or not at all protected) without providing a sufficient logical link between the premise and the conclusion.[34] Although determining the extent to which a work is expressive and/or functional is relevant to the First Amendment's applicability to regulation of that work, it does not end the inquiry.[35] Not all speech is subject to the highest degree of First

[34] The positions taken by the two camps, described in more detail in the remainder of this section, are well-summarized as follows:

> While the government may regulate utilitarian technology free of First Amendment concerns, the First Amendment places limitations on government regulation of speech. If software is found to be speech, then government regulations of software content could trigger strict judicial scrutiny of the regulations. Conversely, if software is found to be a utilitarian product, software regulations are permissible so long as only nominal First Amendment considerations are satisfied. Ultimately, the finding that software is pure speech rather than utilitarian product will often determine whether a government regulation of computer software will withstand constitutional scrutiny.

Halpern, *supra* note 6, at 141.

[35] Universal City Studios, Inc. v. Reimerdes, 82 F. Supp. 2d 211, 220 (S.D.N.Y. 2000), *rev'd sub nom.* Universal City Studios, Inc. v. Corley, 273 F.3d 429 (2d Cir. 2001) ("[T]his Court assumes for

Amendment protection.[36] Defamatory statements and credible death threats are examples of pure speech which receive little, if any, First Amendment protection.[37] Commercial speech receives less protection than political speech but more protection than defamatory speech.[38] Therefore, the mere observation that software is expressive (or even "pure speech") does not mandate that such speech receive full First Amendment protection;[39] rather, it is merely the first step in the First Amendment inquiry.[40] Conversely, the mere fact that a work is "functional" does not mean it cannot receive any First Amendment

purposes of this motion . . . that even the executable [DeCSS] code is sufficiently expressive to merit some constitutional protection. That, however, is only the beginning of the analysis.").

[36] The court in *Chaplinsky* stated:

> There are certain well-defined and narrowly limited classes of speech, the prevention and punishment of which has never been thought to raise any Constitutional problem. These include the lewd and obscene, the profane, the libelous, and the insulting or 'fighting' words•those which by their very utterance inflict injury or tend to incite an immediate breach of the peace.

Chaplinsky v. New Hampshire, 315 U.S. 568, 571–72 (1942). In some cases, there may even be strict liability for harm caused by "pure speech," as in cases where a manufacturer provides inaccurate or defective product information that causes harm to a consumer. *See generally* Nathan D. Leadstrom, *Internet Web Sites as Products Under Strict Products Liability: A Call for an Expanded Definition of Product,* 40 WASHBURN L.J. 532 (2001). *See also* Dan L. Burk, *Patenting Speech*, 79 TEX. L. REV. 99, 122 (2000) (noting that courts generally do not recognize instruction manuals as "defective" except when, as in the case of aeronautical charts, such texts are considered to be "items [that are] 'physically used' in the operation of aircraft, equivalent to a broken compass or inaccurate altimeter.").

[37] For a discussion of defamation and the First Amendment, see RODNEY A. SMOLLA, SMOLLA AND NIMMER ON FREEDOM OF SPEECH § 23 (2002) [hereinafter SMOLLA & NIMMER]. For a discussion of various kinds of threatening speech and the First Amendment, see *id.* §§ 10, 12, and 13. *See generally* Planned Parenthood of the Colombia./Willamette, Inc. v. Am. Coalition of Life Activists, 290 F.3d 1058 (9th Cir. 2002).

[38] SMOLLA & NIMMER, *supra* note 37, § 20.

[39] *See* Orin S. Kerr, *Are We Overprotecting Code? Thoughts on First-Generation Internet Law*, 57 WASH & LEE L. REV. 1287, 1290–93 (2000) (arguing that the mere fact that source code *is* speech should not justify a sweeping rule that it is immune from regulation without reference to what source code *does* in particular circumstances). The court in *United States v. Elcom* stated:

> The mere fact that [computer-implemented speech] occurs at some level through expression does not elevate all such conduct to the highest level of First Amendment protection. Doing so would turn centuries of our law and legal tradition on its head, eviscerating the carefully crafted balance between protecting free speech and permissible governmental regulation.

United States v. Elcom, 203 F. Supp. 2d 1111, 1128 (N.D. Cal. 2002).

[40] For example, the court in *Corley* found that computer source and object code are expressive, but found that the "functionality of computer code properly affects the scope of its First Amendment protection." *Corley*, 273 F.3d at 452. The Amicus Brief for Corley stated:

> The Appellant and its supporting amici have expended substantial rhetorical energy advancing the proposition that DeCSS computer code is expression protected by the First Amendment, as if this observation alone warrants application of strict scrutiny. Yet this is but a truism, an obvious proposition insufficient to decide the case. For *of course* computer code is expression, and *of course* as expression it is protected by the First Amendment. The question is not whether DeCSS computer code is protected by the First Amendment, or whether one applies such labels to it as "functional" or "expressive." The question is whether the First Amendment's protection extends so far as to immunize the Appellants from deliberately marketing such code to facilitate the theft of intellectual property.

Amicus Brief from Law Professors for the Plaintiffs, Universal City Studios, Inc. v. Corley, 273 F.3d 429 (2d Cir. 2001) (No. 00-9185), *available at* 2001 WL 34106441, at *10. *See also* Robert Post, *Encryption Source Code and the First Amendment*, 15 BERKELEY TECH. L.J. 713, 715 (2000) (arguing against Lee Tien's position that the presence of a "speech act" should be sufficient to trigger First Amendment protection on the ground that there are many forms of speech acts, such as product warnings and contracts, which are not and should not be covered by the First Amendment).

protection. A functional work may merit First Amendment protection if it is also expressive,[41] or if it facilitates expression protected by the First Amendment.[42] Attempts to determine whether the functional or expressive aspects of a particular computer program predominate are doomed to fail.[43]

In summary, the function-expression dichotomy is not helpful in the present context because it fails to capture the actual nature of software. In addition, it does not provide a useful mechanism for drawing legal conclusions about the kind and degree of First Amendment protection that source code should receive. A more nuanced approach is warranted if consistency with the nature of software and existing First Amendment jurisprudence is desired.

D. An Alternative Approach

The discussion above demonstrates that the kind and degree of First Amendment protection that should be afforded to source code is not contingent solely upon the classification of software as either speech or a device. In the following sections I propose an alternative analytical approach in which the kind and degree of First Amendment protection afforded to source code in particular cases depends upon the intent of the speaker-programmer and the strength of the causal connection between the speaker-programmer's speech and the alleged harm.[44] This approach incorporates conventional principles of tort law, criminal law, and First Amendment jurisprudence, thereby preserving as much freedom of expression as possible while promoting the legitimate public

[41] The court in *Bernstein I* stated:

> The music inscribed in code on the roll of a player piano is no less protected for being wholly functional. Like source code converted to object code, it "communicates" to and directs the instrument itself, rather than the musician, to produce the music. That does not mean it is not speech. Like music and mathematical equations, computer language is just that, language, and it communicates information either to a computer or to those who can read it.

Bernstein I, 922 F. Supp. 1426, 1435 (N.D. Cal. 1996).

[42] *See* Post, *supra* note 40, at 717 (pointing out that regulation of certain functional devices, such as motion picture projectors, may receive First Amendment scrutiny because of the role that such devices play in facilitating expression that is protected by the First Amendment). The court in *Corley* stated:

> Arguably distinguishing computer programs from conventional language instructions is the fact that programs are executable on a computer. But the fact that a program has the capacity to direct the functioning of a computer does not mean that it lacks the additional capacity to convey information, and it is the conveying of information that renders instructions 'speech' for purposes of the First Amendment.

Corley, 273 F.3d at 447.

[43] John Hart Ely has commented that:

> [B]urning a draft card to express opposition to the draft is an undifferentiated whole, 100% action and 100% expression. It involves no conduct that is not at the same time communication, and no communication that does not result from conduct. Attempts to determine which element "predominates" will therefore inevitably degenerate into question-begging judgments about whether the activity should be protected.

John Hart Ely, *Flag Desecration: A Case Study in the Roles of Categorization and Balancing in First Amendment Analysis,* 88 HARV. L. REV. 1482, 1495 (1975).

[44] *See infra* Parts III, VI.

interest in regulating harm, in addition to avoiding the need to answer the question of whether source code is speech or a device.

I focus on the issues of intent and proximate cause because both would be critical elements in any civil or criminal claim brought against a programmer for harm allegedly caused by the distribution of his or her source code, and because First Amendment jurisprudence takes intent and proximate cause into account when determining whether and to what extent the First Amendment shields a particular defendant-speaker from liability.[45] The underlying reasons for taking intent and proximate cause into account in both common law and constitutional analysis are essentially the same: to ensure that liability only attaches to defendants with a sufficiently culpable intent,[46] and to those whose actions have a sufficiently strong causal connection to the harm in question.[47]

As evidenced by cases such as *New York Times Co. v. Sullivan*,[48] and *Brandenburg v. Ohio*,[49] the First Amendment may impose heightened requirements of intent and causation in common law criminal or tort claims when highly-protected speech forms the factual basis for such claims.[50] I argue that the First Amendment similarly requires that heightened intent and causation requirements be applied in both common law and statutory causes of action involving protected "software speech."[51]

Before reaching this conclusion, however, I also note that the peculiar combination of functionality and expression in computer source code makes "software speech" susceptible to insufficient protection by the First Amendment even if heightened standards of intent and causation are applied.[52] Specifically, I assert that particular source code may not be subject to the highest degree of First Amendment protection, even if such source code qualifies as protected speech.[53] In support of this assertion, it will be demonstrated that difficult and novel questions of harmonizing freedom of expression with harm regulation would remain even in cases that assume the distribution of software constitutes an act of pure speech that is normally protected by the First Amendment.[54] Therefore, I suggest that it is appropriate to consider the use of super-heightened requirements of intent and causation in cases involving software speech, although I also note that such requirements may fail to

[45] See *infra* Parts III.A–B for a more complete elaboration of the relevance of intent and proximate cause to claims brought against programmer-speakers.

[46] For discussion of the role of intent in First Amendment jurisprudence, see SMOLLA & NIMMER, *supra* note 37, §§ 10:21-43.

[47] For discussion of the rigorous causation rule applied in First Amendment cases, see RODNEY A. SMOLLA, FREE SPEECH IN AN OPEN SOCIETY 50–51 (1992).

[48] 376 U.S. 254 (1964).

[49] 395 U.S. 444 (1969).

[50] *See infra* Part III.C.

[51] *See infra* Part VI.B.

[52] *See infra* Parts IV–V.

[53] *See infra* Part IV.G.

[54] *See infra* Part V.

strike the proper balance between freedom of expression and harm regulation.[55] I conclude by describing the practical difficulties that litigants who press for the application of heightened standards should expect to face, and recommend strategies for such litigants to maximize their likelihood of success.[56]

II. THE MULTI-FACED VIRUS AUTHOR

In this section, a series of related hypothetical examples will be presented to illustrate the tension between protection of software speech[57] and the traditional legal regulation of harm by tort law[58] and criminal law. These hypothetical examples will be referred to throughout the subsequent analysis.

In the first hypothetical, a computer programmer develops a computer virus[59] capable of irreversibly erasing large amounts of data from any computer on which it executes, and of propagating itself to other computers over the Internet. The virus is novel and the computer programmer—referred to henceforth as the "virus author"—knows that existing anti-virus software cannot detect the virus or prevent its execution. The virus author has expended a significant amount of creative effort to create the virus, which incorporates several innovative programming techniques. Furthermore, the virus software is written in an interpreted programming language and is therefore executable in source code form.

The virus author distributes the source code over the Internet to one thousand randomly-selected recipients with the sole intent of causing the virus to execute on the recipients' computers and to propagate itself to other computers.[60] To further this goal, the virus author disguises the

[55] *See infra* Part V.

[56] *See infra* Part VI.

[57] The term "software speech" herein refers to speech embodied in software code, whether it be source code or object code. *See* MICROSOFT COMPUTER DICTIONARY, *supra* note 1, at 372 (Object code is "code, generated by a compiler or an assembler, that was translated from the source code of a program."). Whether the term "software speech" should be defined more broadly to encompass any speech that is automatically executable by a computer is beyond the scope of this article.

[58] Although the civil law examples provided herein involve tort law, at least some of the issues raised herein are relevant to other areas of private law, such as contract and intellectual property law.

[59] A computer virus is:

> [a]n intrusive program that infects computer files by inserting in those files copies of itself. The copies are usually executed when the file is loaded into memory, allowing the virus to infect still other files, and so on. Viruses often have damaging side effects – sometimes intentionally, sometimes not. For example, some viruses can destroy a computer's hard disk or take up memory space that could otherwise be used by programs.

MICROSOFT COMPUTER DICTIONARY, *supra* note 1, at 555. In these hypothetical examples the computer virus plays the role of a software program that, when executed, performs a function that may properly be regulated by law. The approach I propose applies equally to other kinds of software, such as software for stealing money from bank accounts, invading privacy, or disabling the security system at a power plant.

[60] Many real-world viruses are distributed in this way. When the hybrid virus-worm CodeRed II arrived at a victim machine (either through an infected email message or through an infected Web

virus as a benign attachment to an email message that cheerily encourages its recipients to open the attachment, most of whom unwittingly take the bait, thereby erasing massive amounts of data, interrupting personal lives and businesses, and causing millions of dollars in losses.

In the second hypothetical, the virus author performs the same actions but with a dual intent: first, to cause the same harm as in the previous hypothetical; and second, to express information through the virus' source code, including the virus author's ideas about computer security in general and the innovative techniques used by the virus to exploit flaws in a widely-used operating system in particular.[61] Assume that the virus' propagation successfully causes the intended harm, expresses the virus author's intended message, and brings about the intended improvements in computer security. Note that the virus author's actions in both this and the previous hypothetical are exactly the same; the only difference between the two cases is the nature of the virus author's intent.[62]

In the third hypothetical, the virus author distributes the same virus with the sole intent of causing harm. This time, however, he does not attempt to disguise the virus as a benign file; rather, he attaches the virus to an email message with a single subject line reading "Warning: Extremely Dangerous Computer Virus Attached." Having been forewarned, none of the recipients unwittingly executes the virus. The virus author intends, however, for at least one of the recipients to choose to propagate the virus maliciously to others. One of the recipients does

server), it would install a Trojan horse to infect the machine and then use the machine's Internet Protocol ("IP") address to generate three hundred to six hundred random IP addresses, which it then attempted to infect. Peter Szor & Eric Chien, *CodeRed II*, Symantec Security Response: CodeRed II, *at* http://www.symantec.com/avcenter/venc/data/codered.ii.html (June 13, 2003).

The "Melissa" virus disguised itself as a friendly email containing a subject line reading "Important message from," followed by the name of someone familiar to the recipient. If the recipient opened the attachment, the virus would forward itself to fifty people in the recipient's address book in the same manner. The spread of the virus clogged networks and brought many systems to a halt. *See* Lisa M. Bowman, *Judge Sends Melissa Creator to Prison*, ZDNET NEWS (May 1, 2002), *at* http://zdnet.com.com/2100-1105-896504.html.

[61] One commentator has distinguished cases involving distribution of computer viruses from cases involving protected software speech by claiming that "releasing a virus or worm onto the Internet lacks any communicative intent, and therefore cannot be a speech act." Tien, *supra* note 6, at 669. Whether a programmer releases a particular virus with communicative intent, however, is a question of fact in each case, as demonstrated by the hypotheticals described herein and the real-world examples cited.

[62] The ideas that may be expressed by distributing a virus are not limited to technological information, but may include political statements such as commentary on social injustice. *See* Kim Zetter, *What Makes Johnny (and Jane) Write Viruses?*, PC WORLD (Nov. 15, 2000), *available at* http://www.pcworld.com/features/article/0,aid,34405,pg,3,00.asp.

> LoveLetter suspect de Guzman was viewed as a hero by fellow students at the AMA Computer College in the Philippines because the Trojan horse he allegedly created was designed to steal Internet passwords. Internet access in the Philippines costs about $90 monthly, a price prohibitive to students in de Guzman's lower-class neighborhood. He was viewed as a hero for robbing from rich ISPs to give to the Internet poor.

Id.

so, transmitting the virus in a deceptive email message to hundreds of other recipients within a few seconds and with a few mouse clicks. Some of the new recipients unwittingly execute the virus, causing it to spread and effectuate the harm described above. Once again, Pandora's box has been opened.

In the fourth hypothetical, the virus author performs exactly the same actions but with the dual intent, and actual effect, of both causing harm and expressing ideas.

In the fifth hypothetical, the virus author has purely benevolent motives. He does not wish to cause anyone harm, but only to express ideas and to improve computer security. This time, the body of the virus-enclosing email message explains that the attachment contains an extremely powerful and novel virus that is capable of causing great destruction, but explicitly instructs against using the virus for malevolent purposes. Furthermore, the email message explains the programming techniques embodied in the virus, the operating system vulnerabilities that it exploits, and the ways in which such vulnerabilities could be fixed. Although the virus has the virus author's intended expressive effects, it results in increased computer security only after a particularly malicious recipient unleashes the virus onto the world, causing all of the damage described above.[63]

[63] It is common for virus authors to submit viruses to antivirus vendors, such as McAfee Security, for the notoriety of having their viruses listed in antivirus software programs. *See id.* Their good intentions can be manipulated by "collectors" who may accept submissions only to release the viruses to cause harm. *Id.*

The case of the "Morris Internet Worm" presents a slightly different fact pattern within the same general paradigm as the fifth hypothetical described above. Robert Tappan Morris, a first-year graduate student in Cornell University's computer science Ph.D. program, developed a computer program that was capable of spreading across the Internet by exploiting computer security flaws to copy itself from one computer to another. United States v. Morris, 928 F.2d 504, 505 (2d Cir. 1991). Morris released the program, which was an example of a "worm," with the intent of "demonstrat[ing] the inadequacies of current security measures on computer networks by exploiting the security defects that Morris had discovered." *Id.* When he released the worm, however, it spread much more quickly than he had anticipated and caused many computers around the country either to crash or to become effectively inoperable. *Id.* at 506. "The estimated cost of dealing with the worm at each installation ranged from $200 to more than $53,000." *Id.* Morris was found guilty, following a jury trial, of violating 18 U.S.C. § 1030(a)(5)(A), also known as the Computer Fraud and Abuse Act ("CFAA"), and was sentenced to three years of probation, 400 hours of community service, a fine of $10,050, and the costs of his supervision. *Id.* His conviction was upheld on appeal. *Id.* at 511. *See also* Thomas Darby & Charles Schmidt, *The Morris Internet Worm*, *at* http://courses.ncsu.edu/classes-a/computer_ethics/abuse/wvt/worm/darby/ (last visited Apr. 15, 2004) (discussing the difference between a worm and a virus).

Another interesting case is that of Simon Vallor, a 22-year old Welsh programmer who was convicted and sentenced to two years in prison for distributing several viruses over the Internet. *Computer Virus Author Jailed*, BBC NEWS (Jan. 21, 2003), *at* http://news.bbc.co.uk/1/hi/wales/2678773.stm. Although the viruses were estimated "to have seized-up 27,000 computers and caused mayhem in 42 countries," *id.*, Vallor claimed that he had not intended to do any harm. *Writer Claims Viruses Were Harmless*, BBC NEWS (Jan. 21, 2003), *at* http://news.bbc.co.uk/1/hi/wales/ 2680419.stm. Rather, he claimed that he wrote the viruses as a programming experiment and did not believe that they would spread far. *Computer Virus Author Jailed, supra.* Furthermore, he claimed that he designed the viruses merely to copy themselves from one computer to another, but not to destroy or

In the sixth and final hypothetical, the virus author, a computer science professor, contacts the manufacturer of the operating system targeted by the virus prior to releasing it and provides the manufacturer with instructions for fixing the security flaws exploited by the virus. The manufacturer makes available a software patch that fixes the problems.[64] The virus author teaches his computer science students about the techniques used by the virus and posts the virus' source code to a Web site intended for use by his students, but which is accessible by the general public. A few weeks later, someone maliciously propagates the virus over the Internet, thereby causing significant damage to a relatively small number of computers to which the operating system patch had not been applied.[65]

III. LEGAL ANALYSIS

A. Intent and Causation

Most people would agree that the virus author in the first hypothetical, who maliciously propagates the virus with no expressive intent, may be held legally accountable for the resulting harm. Most would also agree that the virus author in the last hypothetical, whose intentions are purely benevolent and who takes reasonable steps to limit any potential harmful effects of the virus, should be absolved of any liability.[66] Between these two extremes, however, lies room for much disagreement. My present aim is not to draw substantive conclusions about the virus author's liability in each case, but rather to identify the

corrupt any data. *Id.* The harm caused by the viruses resulted instead from the flood of email messages that were automatically generated to spread the virus. *Id.*

[64] *See* Robert Vamosi, *My Plan for Fixing Software Flaws*, ANCHORDESK (Oct. 1, 2002), *at* http://reviews-zdnet.com.com/4520-6033_16-4207625.html (describing the formation of a committee of five software vendors and security researchers which recommended that: (1) vendors should acknowledge software vulnerabilities sent by security researchers; (2) vendors should follow up with researchers within ten days; (3) vendors should provide updates every seven days; and (4) vendors should resolve the problem within 30 days).

[65] Even when patches are available, and the public is encouraged to fix vulnerable systems, viruses and worms may remain active years after the initial outbreaks. Sam Costello, *Aging Worms Still Crawl, Threaten Net,* IDG NEWS SERVICE (Nov. 24, 2002), *at* http://www.pcworld.com/news/article/0,aid,98504,00.asp. Antivirus vendor Trend Micro reports having had more than 1,500 reports of activity of the "Nimda" worm worldwide in an average 24-hour period a year-and-a-half after the worm's debut. *Id.* The Sapphire worm caused significant disruption to Internet traffic worldwide even though Microsoft had issued a patch to fix the flaw exploited by the worm over six months before the worm was released. Robert Lemos, *Worm Exposes Apathy, Microsoft Flaws*, CNET NEWS.COM (Jan. 26, 2003), *at* http://news.com.com/2100-1001-982135.htm (describing the worm and laying most of the blame for the worm's effects at the feet of system administrators who had failed to apply the patch previously issued by Microsoft).

[66] The last virus author would likely not even need to rely on the First Amendment as an affirmative defense since his actions would likely not satisfy the elements of any civil or criminal claim brought against him.

factors upon which liability may be based and the bearing, if any, that the First Amendment has upon these factors.

I will focus on two factors, the virus author's state of mind[67] and the causal connection between the virus author's actions and the resulting harm.[68] Both factors would be critical elements of essentially any civil or criminal claim that might be brought against the virus author,[69] and both are highly relevant to the First Amendment analysis of such claims.[70] A private party might, for example, bring a common law intentional tort claim, such as trespass to chattels[71] or a negligence claim,[72] against the virus author, depending on the particular facts of the case.[73] Similarly, a criminal claim based on a theory of accomplice liability might be brought against the virus author.[74]

Claims might also be available pursuant to statutes that specifically prohibit causing damage to computer equipment.[75] In each case, it would

[67] I use the terms "state of mind" and "intent" interchangeably herein.

[68] I assume that harm is one that is properly subject to regulation by law. The damage caused by the virus in the hypotheticals described above is an example of such a harm.

[69] By an "intent" requirement, I refer to the state of mind required for a particular element of a claim. The Model Penal Code, for example, defines four mental states, in decreasing order of culpability: purposely, knowingly, recklessly, and negligently. MODEL PENAL CODE § 2.02(2) (2002). Applying this approach to mental state more generally, even criminal and tort causes of action that do not require a "specific intent," but rather require only negligence or impose strict liability, still incorporate an "intent" requirement as that term is used herein.

Both tort and criminal law incorporate a concept of proximate cause, although the meaning of proximate cause differs somewhat from the tort to the criminal law context. *See* JOSHUA DRESSLER, UNDERSTANDING CRIMINAL LAW 157–58 (1990).

[70] Other elements, such as the duty element of a negligence claim, are not discussed here for the sake of brevity, despite the potential relevance of such elements to the issues discussed herein. For a discussion of the difficulty of establishing duty in the virus hypotheticals, see *infra* note 73.

[71] RESTATEMENT (SECOND) OF TORTS § 217 (1965) ("A trespass to a chattel may be committed by intentionally (a) dispossessing another of the chattel, or (b) using or intermeddling with a chattel in the possession of another.").

[72] For a discussion of the elements of a negligence claim, see generally W. PAGE KEETON et al., PROSSER AND KEETON ON TORTS § 30 (5th ed. 1984).

[73] The most difficult hurdle to be overcome by a plaintiff in a negligence suit against any incarnation of the virus author would be establishing that the virus author owed a duty to the plaintiff, and then proving that the virus author breached such a duty. *See generally* Pamela Samuelson, *Can Hackers Be Sued for Damages Caused by Computer Viruses?,* 32 COMM. OF THE ASS'N FOR COMPUTING MACH. INC. 666 (1989). The difficulty in establishing and proving duty and other tort elements in cases such as this was a significant motivation for the enactment of statutes that make abuse of computer equipment actionable without a duty requirement. *See* Mark R. Colombell, *The Legislative Response to the Evolution of Computer Viruses*, 8 RICH. J. L. & TECH. 18 (Spring 2002), *at* http://www.law.richmond.edu/jolt/u8i3/article18.html; Robin A. Brooks, *Deterring the Spread of Viruses Online: Can Tort Law Tighten the 'Net'?*, 17 REV. LITIG. 343, 352 (1998).

[74] For a discussion of the nature of accomplice liability, see DRESSLER, *supra* note 69, § 30.02. The "Hit Man" case is an example of a case in which a publisher was held liable under a theory of aiding and abetting based on the publication of a murder instruction manual. *See infra* note 87 and accompanying text. In general, however, it is unlikely that a mere publisher would be held liable for harm resulting from published software speech. For example, America Online was recently held to be immune from suit by a subscriber who was sent a "punter" (malicious software instructions designed to log the recipient off of the online service) by another subscriber. Green v. Am. Online, 318 F.3d 465, 470–71 (3d Cir. 2003).

[75] The first virus author's liability may be based on any of a variety of tort and criminal causes of action. In addition to the common law claims described below, the virus author may be held liable under various statutes specifically targeting computer-related misconduct. For examples, see CAL.

be necessary to prove that the virus author had the state of mind required by the cause of action and that the virus author's actions constituted the proximate cause of the harm.

B. Role of the First Amendment in Likely Causes of Action

I assume for the purposes of the following discussion that the virus author's expression in the hypotheticals described above constitutes "core protected speech." Note that the term "expression," as used herein, requires the intent to express a message, so that the first and third hypotheticals by definition do not include any expression. In addition, although there is no precise definition for terms such as "core protected speech" or "highly-protected speech," these and similar terms typically refer to speech that conveys political, religious, or scientific messages, particularly those related to matters of public concern.[76] One might think that the assumed protected status of the virus author's speech would guarantee the virus author the ability to raise a successful First Amendment defense against any claim brought against him for harm caused by the virus.[77] The fact that expression is classified as "protected" does not mean, however, that it is shielded absolutely from liability, but rather that stricter legal standards are applied to its regulation in various

PENAL CODE § 502 and W. VA. CODE. § 61-3C-3. States with statutes prohibiting unauthorized access to computers, computer systems, and/or computer data, tend to include viruses in their definitions of "Computer Contaminants." *See, e.g.*, CAL. PENAL CODE § 502(10) (2004); W. VA. CODE § 61-3C-7 (2002). Furthermore, the CFAA prohibits, *inter alia*: (1) intentional unauthorized access that interferes with government operation of government computers; and (2) intentional unauthorized access to a federal interest computer that results in alteration, damage, or destruction of information in the computer or prevents authorized use of the computer or information. 18 U.S.C. § 1030(a)(5) (2000). S*ee generally* Colombell, *supra* note 73; Michael Lee et al., Comment, *Electronic Commerce, Hackers, and the Search for Legitimacy: A Regulatory Proposal*, 14 BERKELEY TECH. L.J. 839 (1999); Michael R. Levinson & Christopher E. Paetsch, *The Computer Fraud and Abuse Act: A Powerful New Way to Protect Information*, 20 INTELL. PROP. L. NEWSL. 24 (2002).

The mere fact that the virus' source code may have some expressive value would not be sufficient to support a First Amendment defense for the virus author in the first hypothetical, given the author's lack of expressive intent and the failure of the virus to actually express a message. *See* Dallas v. Stanglin, 490 U.S. 19, 25 (1989) ("It is possible to find some kernel of expression in almost every activity a person undertakes—for example, walking down the street or meeting one's friends at a shopping mall—but such a kernel is not sufficient to bring the activity within the protection of the First Amendment.").

[76] Political speech is often considered to be at the core of First Amendment protection. *See* Landmark Communications, Inc. v. Virginia, 435 U.S. 829, 838 (1978) ("Whatever differences may exist about interpretations of the First Amendment, there is practically universal agreement that a major purpose of that Amendment was to protect free discussion of governmental affairs.") (quoting Mills v. Alabama, 384 U.S. 214, 216 (1966)). Religious speech is also at the heart of the First Amendment and receives particularly strong protection. *See, e.g.*, Marsh v. Alabama, 326 U.S. 501 (1946); Trinity United Methodist Parish v. Bd. of Educ., 907 F. Supp. 707 (S.D.N.Y. 1995). Scientific speech has also been given relatively broad protection in the interest of promoting the advancement of science, although such speech may be regulated based on national security interests. *See* Allen M. Shinn, Jr., Note, *The First Amendment and the Export Laws: Free Speech on Scientific and Technical Matters*, 58 GEO. WASH. L. REV. 368, 378 (1990). Speech on matters of public concern is given strong protection on the theory that such speech is essential to democratic self-governance. *See* SMOLLA & NIMMER, *supra* note 37, § 2.

[77] *See supra* note 16 and accompanying text.

contexts.[78] For example, a speaker may be held liable, even for engaging in protected speech, if the speech is directed to causing imminent unlawful action and is likely to produce such action.[79] Furthermore, a statute that regulates protected speech may withstand a First Amendment challenge if the statute promotes a compelling government interest using the least restrictive means available.[80] The mere fact, therefore, that the virus author's speech is classified as "pure speech" or even "protected speech" does not necessarily mean that the virus author may not be held liable for harm caused by the virus.

The claims that are most likely to be brought against the virus author are: (1) common law criminal or tort claims such as trespass to chattels[81] and negligence, in the context of which the extent of the First Amendment's applicability is at best unclear; and (2) claims brought pursuant to content-neutral statutes such as those prohibiting property damage, which reviewing courts typically only afford an intermediate level of First Amendment scrutiny.[82] The First Amendment, therefore,

[78] For example, regulation of speech may survive even the "strict scrutiny" standard if the regulation uses the least restrictive means available to further a compelling state interest. *See* United States v. Playboy Entm't Group, Inc., 529 U.S. 803, 813 (2000).

[79] The Supreme Court in *Brandenburg* stated:

> [T]he constitutional guarantees of free speech and free press do not permit a State to forbid or proscribe advocacy of the use of force or of law violation except where such advocacy is directed to inciting or producing imminent lawless action and is likely to incite or produce such action.

Brandenburg v. Ohio, 395 U.S. 444, 447 (1969).

[80] Sable Comm. of Cal., Inc. v. FCC, 492 U.S. 115, 126 (1989) ("The Government may . . . regulate the content of constitutionally protected speech in order to promote a compelling interest if it chooses the least restrictive means to further the articulated interest.").

[81] *See supra* note 71 and accompanying text. Trespass to chattel claims have been brought successfully for harm caused to computer equipment.

> In the present case, plaintiff is physically the recipient of the defendants' messages and is the owner of the property upon which the transgression is occurring . . . . [P]laintiff is not a government agency or state actor which seeks to preempt defendants' ability to communicate but is instead a private actor trying to tailor the nuances of its service to provide the maximum utility to its customers.
>
> Defendants' intentional use of plaintiff's proprietary computer equipment exceeds plaintiff's consent and, indeed, continued after repeated demands that defendants cease. Such use is an actionable trespass to plaintiff's chattel. The First Amendment to the United States Constitution provides no defense for such conduct.

CompuServe Inc. v. Cyber Promotions, Inc., 962 F. Supp. 1015, 1027 (1997). For a critique of the use of trespass to chattels claims in the online context, see generally Dan L. Burk, *The Trouble with Trespass*, 4 J. SMALL & EMERGING BUS. L. 27 (2000).

[82] The Supreme Court in *O'Brien* stated:

> [A content-neutral] government regulation is sufficiently justified if it is within the constitutional power of the Government; if it furthers an important or substantial government interest; if the governmental interest is unrelated to the suppression of free expression; and if the incidental restriction on alleged First Amendment freedoms is no greater than is essential to the furtherance of that interest.

United States v. O'Brien, 391 U.S. 367, 377 (1968). "The distinction between content-based and content-neutral regulations of speech serves as the keystone of First Amendment law." Elena Kagan, *Private Speech, Public Purpose: The Role of Governmental Motive in First Amendment Doctrine*, 63 U. CHI. L. REV. 413, 443 (1996). The term "content-neutral" refers herein both to regulations that apply on their face to speech (such as regulations banning all distribution of leaflets) and to regulations (sometimes referred to as "generally-applicable regulations") that apply on their face to conduct but which may be applied in particular cases to speech (such as the statute in *O'Brien* prohibiting the destruction of draft cards). *See* Geoffrey R. Stone, *Content-Neutral Restrictions*, 54 U. CHI. L. REV. 46, may not shield the virus author from liability even if the virus author's speech is protected speech, depending on the circumstances of the particular case.

With respect to the merits of claims brought against the virus author, a reasonable jury could find that the virus author's speech in at least some of the cases described above satisfies the elements of at least a negligence claim for releasing the virus without exercising due care in light of the reasonably foreseeable effects thereof.[83] Speech may form the factual basis for a negligence claim, as in a case where the defendant supplies false information to another who suffers a pecuniary loss as a result.[84] The speech in such a case does not, however, constitute core protected speech. Therefore, it would be unnecessary to determine whether the First Amendment requires a different analysis or outcome than that required by the common law cause of action itself.

In general, modern cases typically have not involved situations in which core protected scientific or academic speech has satisfied the elements of a generally-applicable[85] common law or statutory claim.[86] Furthermore, modern courts have avoided having to address the impact of the First Amendment on common law claims in several cases involving instructions for performing unlawful acts by refusing to hold that the speech at issue constituted protected speech. Particularly well-known examples of this class of cases include the "Hit Man"[87] and "Soldier of Fortune"[88] cases, in which common law wrongful death and other tort claims were brought against publishers whose publications were allegedly

99 (1987) ("Content-neutral regulations generally fall into one of two categories: either they expressly restrict only communicative activities, or they expressly restrict only noncommunicative activities but they have an incidental effect on free speech."). For a detailed discussion of the doctrine of content neutrality, see SMOLLA & NIMMER, *supra* note 37, § 9.

[83] *See supra* note 72 and accompanying text.

[84] RESTATEMENT (SECOND) OF TORTS § 552 (1977):

> One who, in the course of his business, profession or employment, or in any other transaction in which he has a pecuniary interest, supplies false information for the guidance of others in their business transactions, is subject to liability for pecuniary loss caused to them by their justifiable reliance upon the information, if he fails to exercise reasonable care or competence in obtaining or communicating the information.

Id.

[85] Although the term "generally-applicable" typically refers to statutes and not to common law causes of action, I use the term "generally-applicable" herein to include both statutes and common law causes of action, such as trespass to chattels and negligence, which are not targeted in general at speech. *See, e.g.,* Cohen v. Cowles Media Co., 501 U.S. 663, 670 (1991) (holding that the Minnesota doctrine of promissory estoppel is a law of general applicability because "[i]t does not target or single out the press," but rather "is generally applicable to the daily transactions of all the citizens of Minnesota.").

[86] *United States v. Progressive, Inc.*, 467 F. Supp. 990 (W.D. Wis. 1979), is the most significant, and perhaps the only, counter-example, although the injunction issued in this case was never tested on appeal.

[87] *See* Rice v. Paladin Enters., 128 F.3d 233 (4th Cir. 1997) (commonly referred to as the "Hit Man" case).

[88] *See* Braun v. Soldier of Fortune Magazine, Inc., 968 F.2d 1110 (11th Cir. 1992); Eimann v. Soldier of Fortune Magazine, Inc., 880 F.2d 830 (5th Cir. 1989); Norwood v. Soldier of Fortune Magazine, Inc., 651 F. Supp. 1397 (W.D. Ark. 1987).

responsible for actual and attempted murders carried out by their readers or persons hired thereby.[89] The courts in these cases held either that the speech at issue was not protected speech or avoided answering the question directly, thus making the impact of the First Amendment in such cases at best unclear.

What is lacking, therefore, is a substantial body of cases which address the relevance of the First Amendment to claims that *protected* instructive[90] speech has caused harm to a third party. The question in such a case would be whether the plaintiff may prevail merely upon proving the conventional elements of the common law or statutory claim, or whether the First Amendment requires something more. The answer to this question is critical to determining whether the virus author should be held liable in each of the hypotheticals described above and, more generally, to determining whether programmers should be held liable for the harm caused by the software they create.

[89] In *Norwood*, the plaintiff was allegedly injured due to attempts to injure or murder him and brought suit against the corporation that published "gun for hire" advertisements. *Norwood*, 651 F. Supp. at 1397–98. The court denied a motion for summary judgment on the ground that a reasonable juror could find that the published advertisement had a substantial probability of ultimately causing harm to someone. *Id.* at 1403.

In *Eimann*, the son and mother of a murder victim brought a state wrongful death action against Soldier of Fortune for negligently placing a classified ad by an ex-marine/ex-drill instructor/weapons specialist who advertised the following: "EX-MARINES—67-69 'Nam Vets, Ex-DI, weapons specialist—jungle warfare, pilot, M.E., high risk assignments, U.S. or overseas." *Eimann*, 880 F.2d at 831. The court overturned the $9.4 million verdict, holding that "[w]ithout a more specific indication of illegal intent than [the] ad or its context provided, [Soldier of Fortune] did not violate the required standard of conduct by publishing an ad that later played a role in criminal activity." *Id.* at 838.

In *Braun*, Soldier of Fortune Magazine appealed a $4,375,000 verdict against it in a tort action brought by the sons of a murder victim who had been murdered by a professional contract killer hired from an advertisement placed in Soldier of Fortune. *Braun,* 968 F.2d at 1112. The Court affirmed the verdict because the language in the ad should have "alerted a reasonably prudent publisher to the clearly identifiable unreasonable risk" that the ad was soliciting violent and illegal jobs. *Id*. at 1115. "It follows that a reasonable jury could conclude that the criminal act that harmed appellees was reasonably foreseeable and, accordingly, that the chain of causation was not broken." *Id.*

In *Rice*, the relatives of three murder victims brought a state law wrongful death action against the publishers of a book providing detailed instructions for performing contract murders. *Rice*, 128 F.3d at 241. The plaintiffs alleged that by publishing the book, the defendants aided and abetted the murderer. *Id.* The defendant raised a First Amendment defense. *Id.* The court held that the First Amendment does not protect speech that provides detailed instructions for committing a crime, which is intended to assist its readers in the commission of crimes, and which has no other purpose but to provide such assistance. *Id.* at 255–56. For further discussion of these cases, see generally Neil L. Shapiro & Karl Olson, *Advertiser Liability: Soldier of Fortune Cases Take Deadly Aim at Publishers,* 11 HASTINGS COMM. & ENT. L.J. 383 (1989); Emma Dailey, Rice v. Paladin Enterprises, Inc.*: Does the First Amendment Protect Instruction Manuals on How to Commit Murder?,* 6 VILL. SPORTS & ENT. L.J. 79 (1999); Thomas C. Kates, *Publisher Liability for "Gun for Hire" Advertisements: Responsible Exercise of Free Speech or Self-Censorship?,* 35 WAYNE L. REV. 1203 (1989).

[90] By the term "instructive speech," I refer to speech that instructs the listener to perform certain actions or which explains to the listener how certain actions may be performed.

C. Heightened Intent and Causation Requirements

The First Amendment has been held to impose heightened requirements of intent and causation in certain classes of cases involving highly-protected speech. A "heightened requirement" is one that is more stringent than that required by the underlying claim in the absence of First Amendment considerations. The landmark cases of *New York Times Co. v. Sullivan*[91] and *Brandenburg v. Ohio*[92] are perhaps the two best-known examples of cases in which the First Amendment compelled the application of heightened intent and causation requirements.[93]

In *Sullivan*, the Supreme Court held that the First Amendment imposed a heightened intent requirement in claims of common law defamation involving an issue of public concern directed at a public official because such speech lay at the core of the "marketplace of ideas" that the First Amendment seeks to protect. In particular, the defendant in such a case may only be held liable if his statement was made with "actual malice," which is defined as having been made "with knowledge that it was false or with reckless disregard of whether it was false or not."[94] The actual malice requirement imposed by the First Amendment is a "heightened" intent requirement because it sets a higher threshold

[91] 376 U.S. 254 (1964). In *Sullivan*, one of the three elected Commissioners of the City of Montgomery, Alabama, brought a civil libel action against four individuals and the New York Times Company. *Id.* at 256. Although his name was not mentioned anywhere in the advertisement, the plaintiff alleged that he had been libeled by a full-page advertisement placed by the four individual defendants in the *New York Times*. *Id.* The question before the court was whether the First Amendment barred liability for publication of the advertisement. *Id.* at 268. The Court held that the First Amendment limited the scope of defamation claims arising under state law. *Id.* at 277. In particular, the Supreme Court required that a statement about a public official must be made with "actual malice" to qualify as defamation and that the burden is on the plaintiff to prove actual malice. *Id.* at 279–80. *See infra* note 94 and accompanying text.

[92] 395 U.S. 444 (1969). The appellant in *Brandenburg* was a leader of the Ku Klux Klan and was convicted by the Ohio courts after a television news report was aired broadcasting speeches made by him containing racist statements and intimating that he and others in his organization might be justified in engaging in acts of racial violence in the future. *Id.* at 444–45. The appellant brought a First Amendment challenge to the constitutionality of a state criminal syndicalism statute under which he was convicted. *Id.* at 445. The Court held that states could not forbid "advocacy of the use of force or of law violation except where such advocacy is directed to inciting or producing imminent lawless action and is likely to incite or produce such action." *Id.* at 447.

[93] Heightened intent and causation requirements have been suggested in other contexts based on First Amendment considerations. *See, e.g.,* Justin Myer Lichterman, *True Threats: Evolving Mens Rea Requirements for Violations of 18 U.S.C. § 875(C)*, 22 CARDOZO L. REV. 1961, 1997 (2001) (recommending that federal statutes criminalizing interstate transmission of certain threats be amended to require a *mens rea* of recklessness). One commentator has proposed a more complex scheme to provide heightened protection for media speech that results in physical injury, according to which "a presumption [is created] in favor of constitutional protection for the media, which could be outweighed and rebutted by a consideration of various factors" which the author identifies in detail. Andrew B. Sims, *Tort Liability for Physical Injuries Allegedly Resulting from Media Speech: A Comprehensive First Amendment Approach*, 34 ARIZ. L. REV. 231, 234 (1992). *But see* Steven J. Weingarten, *Tort Liability for Nonlibelous Negligent Statements: First Amendment Considerations*, 93 YALE L.J. 744, 744 (1984) (arguing that the First Amendment should not impose a higher bar to recovery against authors and publishers of non-libelous negligent statements for the physical harm proximately caused by such statements).

[94] *Sullivan*, 376 U.S. at 280.

than that imposed by the common law defamation claim itself (i.e., in the absence of First Amendment considerations).[95]

In *Brandenburg*, the Supreme Court held that states may not prohibit advocacy unless such advocacy is "directed to inciting or producing imminent lawless action and is likely to incite or produce such action."[96] It therefore requires that the causal chain connecting the speaker's speech to the resulting harm be particularly short and direct, imposing a heightened causation requirement in cases involving advocacy of unlawful action.[97] The *Brandenburg* standard has been applied more broadly in various cases brought against speakers for harm that may be caused in the future or harm that has already been allegedly caused by the individual's speech.[98]

IV. Impact on Software Speech

A. The Problem of Heightened Intent and Causation Requirements in Software Speech Cases

I argue that the holdings in *Sullivan* and *Brandenburg* justify the conclusion that the First Amendment requires the imposition of heightened intent and causation standards in both common law and statutory causes of action involving protected software speech.[99] Before reaching that conclusion, however, I argue that the peculiar combination of functionality and expression in computer source code would make software speech susceptible to being under-protected by the First Amendment, even if such heightened intent and causation standards were applied. In the following discussion,[100] I explain why simply adopting the heightened standards fashioned in cases such as *Sullivan* and *Brandenburg* may fail to protect software speech adequately. To remedy this problem, I suggest that "super-heightened" requirements of intent and causation be considered for use in cases involving software

[95] Malice was *presumed* by the state common law. *Id.* at 283. In other words, the plaintiff was not required to prove that the defendant had the requisite intent (malice) to satisfy a claim for general damages under a theory of libel *per se*. *Id.* The defendant could only rebut this presumption by demonstrating that all of the published statements were true. *Id.* The Supreme Court held that this was unconstitutional and required that the plaintiff prove actual malice, thereby imposing a heightened intent requirement in cases involving public officials. *Id.* at 283–84.

[96] *Brandenburg*, 395 U.S. at 447.

[97] Although I will not attempt to define the properties of a causal chain rigorously herein, the "length" and "directness" of a causal chain are related at least to the amount of time that passes between the first and last links in the chain, the number of steps (links) the chain contains, the complexity of the steps, and the nature and degree of the intent necessary to carry out each step.

[98] *See, e.g.,* Hess v. Indiana, 414 U.S. 105, 108 (1973); NAACP v. Claiborne Hardware Co., 458 U.S. 886, 927–28 (1982). Note, however, that the *Brandenburg* standard has not been adopted universally in all cases involving speech, and that there are strong reasons for not adopting it as a universal First Amendment standard. *See infra* note 216 and accompanying text.

[99] *See infra* Part VI.B.2.

[100] *See infra* Parts IV.B–G.

speech, although I caution that such super-heightened requirements may go too far and should, therefore be considered carefully on a case-by-case basis.[101]

B. The ABC's of Software Speech Cases

The present attempt to determine how to translate First Amendment principles into rules that may be applied in software speech cases may be simplified by recognizing that the facts in the virus hypotheticals, as well as in the other examples provided above, fall into a common pattern involving three parties: a speaker *A*, an intermediary (listener) *B*, and a third party *C*. In each case: (1) speaker *A* expresses instructions for performing a set of actions; (2) intermediary *B* receives and reads (or hears) the instructions; and (3) recipient *B* harms third party *C* by carrying out the instructions.[102] In all cases the ultimate legal question is whether speaker *A* may be held liable for the harm caused to third party *C* by intermediary *B*.

I assert that all of the software speech cases brought to date fall within this paradigm, and that software speech cases that arise in the future are likely to continue to fall within this paradigm. The encryption cases,[103] for example, involved export regulations that prohibited the export of encryption technology, including software, without a license. The stated purpose of the encryption export regulations was to promote national security interests by preventing, or at least making more difficult, the use of encryption technology by foreign actors to encrypt messages that could not be decrypted by the U.S. government for national security purposes.[104] The regulations limited the export of encryption software based on a concern that such technologies could be distributed by first person *A* (a programmer) to second person *B* overseas, and then used by *B* to harm the national security interests of third party *C*—the United States. Seen in this way, the application of the regulations to software speech embodied in encryption software source code falls squarely within the *ABC* paradigm set out above.[105]

[101] *See infra* Part V.

[102] *See* Ryan Christopher Fox, Comment, *Old Law and New Technology: The Problem of Computer Code and the First Amendment*, 49 UCLA L. REV. 871, 873 (2002) ("Typically, regulations [of computer code] have targeted a particular class of computer code, usually with the goal of stopping some sort of dangerous activity that the targeted code might facilitate.").

[103] *See generally supra* note 10.

[104] One may disagree with the legitimacy of the regulations' stated purpose or question whether the chosen means were an effective way of accomplishing that purpose. Such disagreements, however valid they may be, do not imply that the regulations' stated purpose was not its actual purpose or that the chosen means did not represent a sincere attempt to advance that purpose.

[105] The encryption export regulations prohibited the speech of programmers *before* such speech had caused any actual harm. For a discussion of the impact of this on the First Amendment analysis, see *infra* Part VI.A.4.

The DMCA prohibits distributing ("trafficking in") devices that circumvent access-control or copy-control measures.[106] The DMCA is intended to address the situation in which first person *A* distributes a circumvention device to intermediary *B*, who uses the circumvention device to cause harm to third party *C*—the holder of the copyright in the work whose access-control or copy-control measures may be the victim of the circumvention device. When applied to speech embodied in source code for circumventing access- or copy-control measures, speaker *A* is the programmer or other person who distributes the source code, intermediary *B* is the recipient of the source code who actually or potentially uses the source code for circumvention, and third party *C* is the copyright holder.[107] The question in such a case is whether speaker-programmer *A* may be held liable for the harm actually or potentially caused to *C* when the intermediary *B* follows the instructions expressed in the speaker's speech by executing the source code. The DMCA therefore also falls within the *ABC* paradigm.

The significance of the fact that software speech cases fit into the *ABC* paradigm is that legal considerations in *ABC*-type cases often differ from legal considerations in other kinds of speech cases. Consider, for example, the class of cases involving speaker *A* whose speech directly harms listener *B* without the need for an intermediary. Examples of such *AB*-type cases include cases involving regulation of obscene or indecent speech.[108] In such cases, it is the speech itself, rather than any action taken in response to the speech, that is alleged to cause psychological harm. The legal question in such *AB*-type cases is whether the speaker may be held directly liable for the harm caused by his or her speech.

The alleged harm in *AB*-type cases typically is psychological because such cases are, by their nature, concerned only with harms that can be caused directly by speech. For example, the harm that purportedly justifies regulation of obscene speech is, at most, a psychological harm to those who are confronted with obscene materials and find such materials offensive.[109] Although *ABC*-type cases may also involve psychological harms, such cases more generally may involve

[106] 17 U.S.C. § 1201 (2000).

[107] The DMCA prohibits the distribution of anti-circumvention devices, even before such devices have been used to cause any harm. For purposes of the present analysis, however, I treat this prospective feature of the encryption export regulations and the DMCA as irrelevant. For a more detailed discussion of the prospective regulation of the distribution of software, see *infra* Part VI.A.4.

[108] For an overview of the history and current status of the First Amendment's treatment of obscene and indecent speech, *see generally* SMOLLA & NIMMER, *supra* note 37, § 14.

[109] *See* Miller v. California, 413 U.S. 15, 24 (1973) (holding that obscene speech is unprotected by the First Amendment, and defining obscene works as speech works that: (a) "the average person, applying contemporary community standards," would find, when taken as a whole, to appeal to the prurient interest; (b) "depicts or describes, in a patently offensive way, sexual conduct specifically defined by the applicable state law;" and (c) "taken as a whole, lack[] serious literary, artistic, political or scientific value." According to this definition, obscene works are not alleged to cause any physical harm. Rather, the patent offensiveness of such works presumably causes individual psychological harm or harm to community norms.).

physical[110] or pecuniary[111] harms allegedly caused by intermediary *B* to third party *C* in response to the speech of speaker *A*. Furthermore, because *ABC*-type cases involve not only the speech of speaker *A*, but also an action taken by intermediary *B* that harms a legally-protected interest of third party *C*, such cases are likely to involve common law or statutory causes of action that target the action taken by *B* and which, therefore, only incidentally implicate the speech of speaker *A*.[112]

Because of these differences between *AB*-type cases and *ABC*-type cases, one should proceed with caution before attempting to apply principles developed in *AB*-type cases directly to *ABC*-type cases. For example, although the Supreme Court in *Reno v. ACLU*[113] stated that "our cases provide no basis for qualifying the level of First Amendment scrutiny that should be applied to" the Internet,[114] *Reno* was an *AB*-type case involving a challenge to a statute (the Communications Decency Act) which regulated indecent speech on its face.[115] One must therefore be careful before applying the reasoning in *Reno* to *ABC*-type software speech cases and before interpreting *Reno* to hold that speech receives the highest level of First Amendment scrutiny solely by virtue of being disseminated over the Internet. Such attempts to interpret *Reno* broadly,[116] irrespective of the nature of the claim brought against such speech, confuse more than they illuminate. In the following sections, additional ways are described in which existing precedents in the area of software speech fail to provide clear guidance for deciding the increasing number and variety of software speech cases likely to arise in the future.

C. Existing Precedents Provide Poor Vehicles for Focused Analysis

Although the encryption export regulations and the DMCA have been the basis of almost every software speech case decided to date and have been the subject of extensive litigation and commentary, they are far from ideal vehicles for analyzing the fundamental questions of First Amendment law raised by software speech. For example, both the encryption export regulations and the DMCA are objectionable on multiple grounds, some of which are speech-related but not related to the speech of programmers, and some of which are not speech-related at all.

[110] *See, e.g., supra* note 89.

[111] *See supra* note 12 and accompanying text.

[112] *See infra* Part VI.A.2.

[113] 521 U.S. 844 (1997).

[114] *Id.* at 870.

[115] *Id.* at 859–61.

[116] *See, e.g.,* Brief for Defendant-Appellant, at Part IV.C, Universal City Studios, Inc. v. Corley, 273 F.3d 429 (2d Cir. 2001) (No. 00-9185), *available at* http://www.eff.org/IP/Video/MPAA_DVD_cases/20010119_ny_eff_appeal_brief.html (last visited Apr. 15, 2004) (arguing that the Internet, according to *Reno*, is a "fully protected medium" and citing *Reno* as authority for the proposition that § 1201 of the DMCA must be subjected to strict scrutiny).

Courts and commentators often have conflated these different objections, further muddying the waters, as will be described below.

1. Existing Precedents Involve Both Speech of Programmers and Speech of Programs' Users

In particular, First Amendment analysis of the encryption export regulations and the DMCA is complicated by the fact that both target technologies that are used for speech-related purposes. The encryption export regulations target encryption technology used to encrypt messages and thereby to facilitate private and secure communications. The DMCA targets devices that may be used to access or copy audiovisual works such as movies, songs, and other works of expression. Therefore, both the encryption export regulations and the DMCA at least potentially implicate the First Amendment rights of the *users* of the regulated technology, and both have been challenged on this basis.[117] In this sense, the encryption export regulations and the DMCA are akin to laws regulating the distribution of pencils, typewriters, paper, and other products that facilitate expression but which are not themselves forms of expression.[118]

The DMCA, for example, may limit the extent to which a professor may *use* circumvention software to copy a portion of a movie stored on a DVD and display that portion in a classroom for academic purposes, thereby engaging in a fair use of the movie. Whether the DMCA violates the First Amendment rights of the *professor* in such a case is analytically distinct from the question of whether prohibiting distribution of the circumvention software used by the professor is a violation of the First Amendment rights of the software's *programmer*. The merits of the fair use objection would, for example, be unaffected if the issue were instead whether the professor could use circumvention *hardware* for the same purposes. Conversely, the merits of the programmer's First Amendment challenge would be unaffected if the software were instead used for performing a function unrelated to expression, such as software for guiding a missile. The fact that the encryption export regulations and DMCA present potential violations of the First Amendment rights of both *programmers* and of the people who *use* their programs makes the

[117] *See, e.g., Bernstein II*, 945 F. Supp. 1279, 1287 (N.D. Cal. 1996) ("[P]laintiff advances the novel proposition that the First Amendment also includes the right to speak confidentially, and thus, encryption is deserving of protection because it facilitates private communication."); United States v. Elcom, Ltd., 203 F. Supp. 2d 1111, 1122 (N.D. Cal. 2002) (noting defendant's argument that the DMCA "impermissibly infringes upon the First Amendment rights of third parties to engage in fair use"); Brief for the Appellee, *Bernstein IV* (No. 97-16686), *supra* note 6, at 26 (arguing that the use of encryption software to "make speech private" is protected by the First Amendment).

[118] *See* Post, *supra* note 40, at 717 (pointing out that regulation of certain functional devices, such as motion picture projectors, may receive First Amendment scrutiny because of the role that such devices play in facilitating expression that is protected by the First Amendment).

First Amendment analysis rife with confusion when attempting to analyze only the rights of programmers.[119]

2. Existing Precedents Involve Regulations that Implement Controversial Policies

The encryption export regulations and DMCA cases are also less than ideal for discretely analyzing the applicability of the First Amendment to software speech because both implement policies that are controversial for reasons unrelated to the speech of programmers. In particular, critics of the encryption export regulations and the DMCA have argued that the particular measures employed by the regulations are not effective ways of achieving their stated policy ends.[120] Furthermore, both the encryption export regulations and the DMCA address harms that are *malum prohibitum* as opposed to *malum in se*. Indeed, both define and target newly-defined harms that many have argued should not even be considered *malum prohibitum*.[121]

The only laws that programmers have challenged on First Amendment grounds—the encryption export regulations and the DMCA—are objectionable on other grounds as well, and regulate harms that many believe are not particularly harmful. It is politically easy to

[119] The district court in the *Bernstein II* case, for example, correctly recognized this distinction. *Bernstein II*, 945 F. Supp. at 1287 ("Snuffle is speech afforded the full protection of the First Amendment not because it enables encryption, but because it is itself speech.").

[120] *See* Brief for the Appellee, *Bernstein IV* (No. 97-16686), *supra* note 6 ("[T]he regulations are so clumsily written that they do not even achieve this end [of preventing foreign intelligence targets from obtaining encryption technology]."); McClure, *supra* note 10, at 474 (citing the argument made by opponents of the encryption export regulations that "[a]nyone who wants strong encryption can already get it" and that the regulations are therefore incapable of achieving their stated goals).

[121] The encryption export regulations were motivated by the desire to keep strong encryption technologies out of the hands of foreign governments and individuals and thereby decrease the ability of such actors to wage "information warfare" on the United States or evade eavesdropping by the United States. McClure, *supra* note 10, at 472–73. Critics of the encryption export regulations often point to the beneficial uses of encryption to argue that use and distribution of encryption technologies should not be regulated. *See generally* Brief for the Appellee, *Bernstein IV* (No. 97-16686), *supra* note 6; John A. Fraser, III, *The Use of Encrypted, Coded and Secret Communications is an "Ancient Liberty" Protected by the United States Constitution*, 2 VA. J. L. & TECH. 2 (1997).

The DMCA was motivated by at least the government interests in preventing unauthorized copying of copyrighted works and in promoting electronic commerce. *Elcom*, 203 F. Supp. 2d at 1124. Many have argued that copyright holders do not experience any harm when their copyrighted works are accessed for purposes of engaging in fair use, and that the DMCA's relatively strict prohibition against circumventing access- and copy-controls without sufficient exceptions for fair use are therefore unjustified. *See, e.g.,* Fred von Lohmann, *The DMCA Goes Too Far*, NETWORK WORLD (Dec. 10, 2001), *at* http://www.nwfusion.com/forum/2001/1210faceoffyes.html; Lawrence Lessig, *Jail Time in the Digital Age*, N.Y. TIMES, July 30, 2001, at A17; ELECTRONIC FRONTIER FOUNDATION, *Unintended Consequences: Three Years Under the DMCA* (May 3, 2002), *at* http://www.eff.org/IP/DMCA/20020503_dmca_consequences.html. Furthermore, many have argued that copyright holders do not experience any harm when circumvention devices are merely *distributed* rather than *used*, and that the DMCA's prohibition against "trafficking" in circumvention devices is therefore unwarranted. *Id.* The defendants in *Elcom*, for example, argued that the Intellectual Property Clause of the U.S. Constitution precluded Congress from protecting intellectual property rights in the manner embodied in the DMCA. *Elcom*, 203 F. Supp. 2d at 1138. The court in *Elcom* rejected this challenge. *Id.* at 1128.

challenge such laws because the underlying policies they implement are themselves of questionable validity, whether or not implementing such policies involves the regulation of software speech. One must wonder whether First Amendment objections would carry the same weight when applied to statutes, such as those prohibiting theft or destruction of computer equipment, that regulate actions which are uncontroversially *malum in se*.[122]

Furthermore, the combination of the fact that existing precedents have involved software that is *used* for expressive purposes, and the fact that such precedents have involved regulations that promote controversial policies in controversial manners, tends to bias the debate in favor of stronger First Amendment protection for software. In a sense, the cases to date do not necessarily constitute a representative sample of all possible software speech cases, but rather, constitute a sample of cases in which strong First Amendment protection may appear desirable primarily because they have involved programmers who garner sympathy for having written socially-beneficial software. We should be hesitant to formulate broad legal rules based on such cases for the same reason that we should be hesitant to formulate broad legal rules based solely on cases involving conventional speech[123] that is neither harmful nor offensive.

None of the cases decided to date demonstrates that source code *in general*—rather than source code that is particularly expressive or that does not cause harm—should receive strong First Amendment protection as a distinct category of expression. Indeed, to establish a bright-line rule that the expressiveness of source code mandates strong First Amendment protection for software speech in general, regardless of any harm that it causes or may cause, would be to provide much stronger

[122] To the best of this author's knowledge, the only other reported case in which a court has addressed the question of whether source code is protected speech is *United States v. Mendelsohn*, which involved a criminal prosecution against distributors of a software program designed to assist in illegal gambling operations. 896 F.2d 1183, 1184 (9th Cir. 1990). The distributors were convicted for conspiring to transport and aiding and abetting the interstate transportation of wagering paraphernalia, in violation of 18 U.S.C. § 371 (1953). *Id.* The defendants appealed, arguing that the software was speech protected by the First Amendment, comparing the software to an instruction manual for a computer. *Id*. The Ninth Circuit upheld the defendants' convictions based on the defendants' criminal intent and lack of expressive intent, finding that "[t]here is no evidence in this case that any speech by Defendants was directed to ideas or consequences other than the commission of a criminal act." *Id.* at 1185. Interestingly, the court appeared willing to accept, for purposes of argument, that the program was a kind of speech, but reasoned that the relevant question was "whether [the software] is protected speech." *Id*. The court held that the software was not protected speech because:

> [it] is too instrumental in and intertwined with the performance of criminal activity to retain first amendment protection. No first amendment defense need be permitted when words are more than mere advocacy, "so close in time and purpose to a substantive evil as to become part of the crime itself." We conclude that the . . . computer program [at issue] was just such an integral and essential part of ongoing criminal activity.

Id. at 1186 (citation omitted).

[123] By the term "conventional speech," I refer to speech expressed in a natural language, which provides instructions to be followed without using software.

First Amendment protection for source code than for conventional speech.[124]

Modern First Amendment jurisprudence has developed primarily through cases involving particularly harmful, offensive, and unpopular speech, such as racist speech[125] and speech providing instructions for performing an action that is uncontroversially *malum in se*.[126] The software equivalents of such cases would provide better vehicles through which to develop rules for applying the First Amendment to software speech, because such cases truly force the question of whether the First Amendment requires that heightened standards be applied to harm-regulating laws when such laws are applied to software speech.

D. Considering Ideal Cases

In summary, what is lacking are cases involving laws targeted at behavior that is uncontroversially *malum prohibitum* or *malum in se* and which are applied to software that is not used for speech-related purposes when it is executed. The virus hypotheticals described above satisfy these criteria because they involve causes of action, such as common law trespass to chattels and negligence, as well as statutes

[124] *See* Kerr, *supra* note 39, at 1291–92.

> According to Judge Martin [in *Junger*], the "expression" that source code communicates is information about the source code itself, and Professor Junger's encryption source code was expressive because it provided a means of sharing ideas about how to author encryption source code. But according to this reasoning, every series of computer instructions warrants First Amendment protection because code will always convey information about itself. The source code of a program that does *X* will *always* be the author's means of expressing how to write a program that does *X*, no matter what *X* actually is. For example, the source code of the destructive Love Bug computer virus that infected computers worldwide in May 2000 was the author's means of expressing how to write a particularly destructive computer virus. No matter what the source code actually is or does, *Junger* indicates, its status as source code automatically entitles the code to First Amendment protection. No exceptions.
>
> . . . [This approach] treats cyberspace radically differently from the physical world. The problem is that *everything* is "an expressive means for the exchange of information and ideas" about itself, and this is just as true in realspace as in cyberspace. For example, imagine that you have designed a new kind of padlock, and you wish to explain to me how the lock works. The best way to communicate that set of ideas is to give me one of the locks, let me play with it, take it apart, and see for myself how it operates. Sure, you could write a book that offers "a prose explanation" for how the padlock works, but I will learn much more by examining the lock first-hand. To borrow a phrase from the *Junger* court, access to the lock itself provides "the most efficient and precise means by which to communicate ideas" about it. And so it is for everything else in the world. Robbing a bank provides the most instructive way to teach someone how to rob a bank; kicking someone in the shins provides an excellent way of communicating the concept of kicking someone in the shins. So long as the only "expression" we are concerned with is information about the act or thing itself, that act or thing is bound to be "an expressive means for the exchange of information and ideas" about it.
>
> But does this mean that you have a First Amendment right to distribute padlocks, to rob banks, and to kick people in the shins? Of course not. In the physical world, we recognize that the "expression" that the First Amendment protects goes beyond what things *are* to what they *say*.

Id.

[125] *See, e.g.,* R.A.V. v. City of St. Paul, 505 U.S. 377 (1992); Brandenburg v. Ohio, 395 U.S. 444 (1969).

[126] *See, e.g.,* Rice v. Paladin Enters., 128 F.3d 233, 242–43 (4th Cir. 1997); United States v. Progressive, Inc., 467 F. Supp. 990, 992–93 (W.D. Wis. 1979).

prohibiting damage to computer equipment, that are uncontroversial in that they are targeted at behavior that is *malum in se* and involve software—a computer virus—that is not used for speech-related purposes when it is executed. Such cases provide a purer vehicle for analysis because they present the issue of the programmer's First Amendment rights squarely and without the complications introduced by other valid, but irrelevant, concerns.

The virus hypotheticals are also good vehicles for focusing the discussion because the relevant causes of action, particularly the common law causes of action, were clearly not created to target the speech of programmers. The English common law courts were not trying to suppress the speech of programmers when they developed trespass to chattels and negligence. The encryption export regulations and the DMCA, in contrast, have been attacked on the grounds that the executive branch[127] and Congress, respectively, were targeting the speech of programmers when they enacted those laws.[128] The courts that have addressed this question have failed to answer it consistently,[129] and even

[127] Although administration of the encryption export regulations was originally within the jurisdiction of the State Department, President Clinton transferred jurisdiction to the Department of Commerce by executive order in 1996. *See Bernstein III*, *supra* note 10, at 1291.

[128] *See, e.g.,* Brief for the Appellee, at Part V.A.2, *Bernstein IV* (No. 97-16686), *supra* note 6 (arguing that "The cryptography regulations are a censorship scheme that must receive the strictest judicial scrutiny, not the attenuated review proposed by the government.").

[129] *See* Universal City Studios, Inc. v. Reimerdes, 111 F. Supp. 2d 294, 329 (S.D.N.Y. 2000) ("The reason that Congress enacted the anti-trafficking provision of the DMCA had nothing to do with suppressing particular ideas of computer programmers and everything to do with functionality."); The court in *Elcom* stated:

> On its face, the [DMCA] does not target speech. Section 1201(b) bans trafficking in devices, whether software, hardware, or other. Thus, strict scrutiny is not appropriate in the absence of any suggestion that Congress sought to ban particular speech, qua speech. Courts that have considered the issue in the context of the DMCA have determined that Congress was not concerned with suppressing ideas but instead enacted the anti-trafficking measures because of the function performed by the code.

United States v. Elcom, 203 F. Supp. 2d 1111, 1128 (N.D. Cal. 2002); Universal City Studios, Inc. v. Corley, 273 F.3d 429, 454 (2d Cir. 2001) (holding that the DMCA is a content-neutral restriction regulating only the non-speech portions of code that is used to circumvent technologies that limit access to copying. "The DMCA, as applied to the Appellants, and the posting prohibition of the injunction target only the nonspeech component."); *Elcom*, 203 F.Supp. 2d at 1129 ("Accordingly, the court concludes that intermediate scrutiny, rather than strict scrutiny, is the appropriate standard to apply.").

Some litigants have argued, unsuccessfully, that the DMCA specifically targets speech and should thus be subject to strict scrutiny. *See, e.g.,* Plaintiffs' Brief in Opposition to RIAA, SMDI and Verance's Motion to Dismiss, at Part I.A.3, Felten v. RIAA (D.N.J. Aug. 13, 2001) (No. CV-01-2669 (GEB)) *available at* http://www.eff.org/IP/DMCA/Felten_v_RIAA/20010813_eff_felten_brief.html. ("The DMCA targets anyone who published information that can be used to circumvent copyright control measures, regardless of whether the person is a university scientist or not. . . . These provisions, as the Defendants read them, apply to the publication and dissemination of scientific research."). *See also* EFF Supplemental Letter Brief, *Universal City Studios*, 273 F.3d 429 (No. 00-9185), *available at* http://www.eff.org/IP/VIDEO/MPAA_DVD_cases/20010530_ny_eff_supl_brief.html (last visited Apr. 15, 2004) ("As applied here to computer code, the anti-trafficking provisions [of the DMCA] are content-based in that they specifically target the publication of, and linking to, scientific expression based on the particular topic addressed by that expression – namely, techniques for circumventing CSS."). Brief of Amici Curiae Dr. Harold Abelson et al., at Part V, *Reimerdes* (No. 02-9185), *supra* attempting to determine the actual intent of the State Department or Congress is to go afield of the central problem addressed herein.

For clarity of analysis it is more instructive to consider cases in which legislative intent is pure and not directed at the suppression of speech. Even such cases indicate that software raises new, interesting, and extremely difficult problems for First Amendment jurisprudence. In addition to the virus hypotheticals described above, for example, one might imagine any situation in which the federal government has a legitimate interest in prohibiting or regulating the mere distribution or possession of particular classes of technological devices or substances, such as bomb-making materials,[130] nuclear fissile material,[131] or computer viruses, because of the high risk of significant harm resulting from their use. Such regulations may be motivated not by a desire to suppress speech, but rather by a desire to prevent harm caused by the regulated devices. Prior to the advent of software, cases in which such regulations were applied to speech were the exception rather than the rule. Now, however, some such regulations, such as those regulating the distribution of computer viruses, may apply solely or primarily to speech embodied in software, even though the regulations are not specifically directed at speech. This type of situation raises the difficult question of whether the First Amendment requires such regulations to be nullified, or at least to

note 5 (arguing that regulation of computer code, whether in source code or object code form, should be subjected to strict scrutiny).

The court in *Bernstein II* held that, although "[t]he bulk of the ITAR scheme may well be viewed as a law of general applicability not aimed at expression . . ., the same cannot be said of Category XIII(b) of the Munitions List. Category XIII(b) is directed very specifically at applied scientific research and speech on the topic of encryption." *Bernstein II*, 945 F. Supp. 1279, 1288 (N.D. Cal. 1996). *See also Bernstein III*, 974 F. Supp. 1288, 1306 (N.D. Cal. 1997) ("The encryption regulations, like Category XIII of the USML, is specifically directed at speech protected by the First Amendment.").

[130] DEL. CODE ANN. tit. 11, § 1338 (1973) ("Whoever manufactures, transfers, uses, possesses or transports any bomb, incendiary device, Molotov cocktail or device designed to explode or produce uncontained combustion with intent to cause bodily harm or damage to any property or thing shall be guilty of a class D felony."). In contrast, the court in *Karn* held that:

> [T]he government regulation at issue here is clearly content-neutral. . . . The defendants are not regulating the export of the diskette because of the expressive content of the comments and/or source code, but instead are regulating because of the belief that the combination of encryption source code on machine readable media will make it easier for foreign intelligence sources to encode their communications.

Karn v. U.S. Dept. of State, 925 F. Supp. 2d 1, 30 (D.D.C. 1996). Similarly, the district court in *Junger* held that the encryption export regulations were content-neutral and therefore subject to intermediate scrutiny. Junger v. Daley, 8 F. Supp. 2d 708, 720 (N.D. Ohio 1998). The Court of Appeals in *Junger* sidestepped the issue of congressional intent and simply held that source code is protected by the First Amendment because "source code is an expressive means for the exchange of information and ideas about computer programming." Junger v. Daley, 209 F.3d 481, 485 (6th Cir. 2000).

[131] CAL. VEH. CODE § 14611:

> No person shall knowingly direct the operation of a vehicle transporting fissile class III shipments or large quantity radioactive materials, as defined in Section 173.389 of Title 49 of the Code of Federal Regulations, by an individual who does not possess a license of the appropriate class with a radioactive materials driver's certificate authorizing that transportation attached to the license.
>
> A person convicted under this section shall be punished by a fine of not less than five thousand dollars ($5,000) nor more than ten thousand dollars ($10,000).

Id.

satisfy a heightened standard of scrutiny, when applied to software speech.

A clearer example is that of a statute which prohibits theft. Such a statute clearly does not target, nor does it intend to suppress, speech. Furthermore, such a statute does not even address, on its face, actions committed when using computer software. Now imagine that a programmer writes a computer program that is capable of electronically extracting money from any bank account held at a major bank, and the programmer then distributes the software in the various ways described above in the virus hypotheticals. The programmer may be held liable for theft for aiding and abetting in at least some cases.[132] In situations in which the programmer distributed the software with at least some expressive intent, the statute on its face would apply to speech, and the applicability of the First Amendment would therefore be at issue.

One might think that the situations just described are the exceptions rather than the rule and are not likely to occur as a practical matter. To the contrary, I assert that the DMCA may be understood as a generally-applicable regulation that is not targeted at the speech of programmers, but which may apply incidentally to such speech.[133] Furthermore, generally-applicable laws are likely to increasingly run up against the First Amendment rights of programmers, as the functions performed by software become more diverse and powerful, as software becomes more widely used, and as the potential for software to cause harm increases.[134] Statutes prohibiting identity theft, tampering with power plants, and embezzlement[135] all prohibit actions that may now be performed using software. Clearly, such statutes are neither facially unconstitutional nor are they intended to suppress the speech of programmers merely because they are capable of being applied to software whose source code is expressive. Furthermore, as the virus hypotheticals illustrate, the First Amendment cannot impose a *per se* bar on applying such statutes to expressive source code if the use of such source code satisfies the elements of the statute. If a *per se* bar against application of regulations of this type to distribution of source code is rejected, the problem must be addressed by developing clear rules for defining the circumstances in which such distribution may be regulated, so as to provide maximum protection for freedom of expression, while allowing Congress to regulate harms that are within its power to regulate.

[132] *See generally* United States v. Mendelsohn, 896 F.2d 1183 (9th Cir. 1990).

[133] *See* Universal City Studios, Inc. v. Corley, 273 F.3d 429, 454 (2d Cir. 2001) ("The Appellants' argument fails to recognize that the target of the posting provisions of the injunction—DeCSS—has both a nonspeech and a speech component, and that the DMCA, as applied to the Appellants, and the posting prohibition of the injunction target only the nonspeech component.").

[134] *See infra* Part VI.A.2.

[135] *See, e.g.,* CONN. GEN. STAT. § 53a-119 ("A person commits embezzlement when he wrongfully appropriates to himself or to another property of another in his care or custody."); 22 USCA § 4217 (1990) ("[Any consular officer] deemed guilty of embezzlement, and shall be punishable by imprisonment for not more than five years, and by a fine of not more than $2,000.").

E. The Causal Power of Software

In this and the following sections, I propose an alternative analytical framework to be applied in software speech cases. The framework that I propose is premised on the fact that software speech is "powerful" speech in comparison to conventional speech, in a way that is relevant to the legal analysis. In this section, I provide a justification for this claim that software speech is powerful speech.

Even if software speech is pure, core, protected speech, it is speech with particularly potent causal powers. Software can guide missiles and airplanes, steal money from bank accounts and protect bank accounts against theft, detect tumors in x-rays, and control robots that build other machines. Even if we assume that source code is speech, it is speech that is capable of directly and automatically controlling computers and machines connected to computers.[136] In other words, it is speech that can directly and automatically cause machines to perform actions.[137] I therefore refer to software speech as a kind of "functional expression."

One might wonder how the causal powers of software speech differ from the causal powers of other kinds of speech which direct people to perform actions or explain how machines operate. The feature of software speech that distinguishes it from other kinds of speech is that it is capable of controlling a machine *directly*, without the need for a human intermediary.[138] A military general, for example, who issues

[136] *See* Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2317–29 (1994):

> To put the point starkly: no one would want to buy a program that did not *behave*, i.e., that did nothing, no matter how elegant the source code "prose" expressing that nothing . . . The goal of a programmer designing software is to achieve functional results in an efficient way. While there may be elements of individual style present in program design, even those style elements concern issues of industrial design, e.g., the choice of one or another programming technique or the clarity (or obscurity) of the functional purpose of a portion of the program.

Id.

[137] By asserting that a particular software program "can" cause a machine to perform an action, I merely assert that the software program is capable of being used in a manner which causes the machine to perform the action. This should address any potential objection that software cannot, by itself, cause a machine to perform an action unless and until a person uses the software to cause the machine to perform the action. In the same way, it is accurate to assert, for example, that a gun "can" kill a person even if the gun is only capable of killing someone when used in a particular way by a person.

One commentator has attempted to equate source code with other kinds of speech by arguing that "[i]f someone publishes bomb-making information, it only has effects if someone else uses it to make a bomb. [It is true for all kinds of speech that] [t]he effects always run through a person's use of it." Tien, *supra* note 6, at 693. The problem with this argument is that it explains too much, because it is also true that bombs themselves "only [have] effects if someone else uses [them]," *id.,* and this fact does not render the law powerless to regulate the use or even the distribution of bombs. The criticism just cited begs the question as to what rules should be applied to determine whether particular laws regulating the distribution of a particular kind of object are permissible, particularly if the object and/or the act of distributing it is expressive. Establishing such rules is the difficult problem with which this paper grapples.

[138] The court in *Corley* stated:

> Unlike a blueprint or a recipe, which cannot yield any functional result without human comprehension of its content, human decision-making, and human action, computer code can instantly cause a computer to accomplish tasks and instantly render the results of those tasks

commands to his troops, may cause them to take the actions dictated by his instructions, but in such a case there are human intermediaries—the troops—interposed in the causal chain between the military general and the performance of the actions he orders to be performed. The same is true of a corporate CEO who instructs her company to implement a new business strategy or a parent who instructs a child to sit still.

Now consider an explanation of how a machine operates, such as a schematic diagram for a circuit or a natural language description of the internal operation of an automobile. The mere act of expressing such explanations, whether orally or in writing, is not sufficient to cause the circuit or automobile to be built or to cause an existing circuit or automobile to perform the functions described by the explanation.[139] In either case, it is necessary for a human to read the explanation, construct a working machine based on the explanation,[140] and activate the machine. The mere expression of the explanation cannot cause the functions it describes to be performed in the absence of a human intermediary.

Furthermore, in many, if not most, cases it is necessary for the human intermediary to exercise discretion in interpreting the speaker's expressed instructions and to take actions necessary to implement those instructions. For example, in the case of written instructions explaining the internal operation of an automobile, it may be necessary for the reader who wishes to construct an automobile based on such an explanation to exercise a significant amount of judgment, skill, and perhaps even creativity, to interpret the instructions, select appropriate equipment and raw materials, and construct a working automobile. Executing the instructions may further involve the exertion of a significant amount of physical effort by the reader over an extended period of time. The causal chain linking the speaker (the instructions' author) to the constructed automobile is attenuated by the physical effort and extent of discretion exerted by the reader, and the amount of time

> available throughout the world via the Internet. The only human action required to achieve these results can be as limited and instantaneous as a single click of a mouse. These realities of what code is and what its normal functions are require a First Amendment analysis that treats code as combining nonspeech and speech elements, i.e., functional and expressive elements.

Corley, 273 F.3d 429, 451 (2d Cir. 2001).

[139] It is now possible, however, to design and implement electronic circuitry based on descriptions of such circuitry which resemble computer programs. Electrical engineers may now design certain kinds of circuits in purely functional terms using hardware description languages such as Verilog and Verilog Hardware Description Language. *See also* James R. Goodman et al., *Toward a Fact-Based Standard for Determining Whether Programmed Computers are Patentable Subject Matter: The Scientific Wisdom of Alappat and Ignorance of Trovato*, 77 J. PAT. & TRADEMARK OFF. SOC'Y 353, 357–60 (1995) (describing commercially available tools that allow electrical engineers to design circuits in terms of the functions they perform). Such a description may be provided to a "compiler," which generates a description of the physical structure of a circuit that performs the specified functions. This description may be provided directly to a foundry to fabricate the physical circuit itself. *See generally* Verilog Resources, *at* http://www.verilog.com/ (last visited Apr. 15, 2004); *EDA Industry Working Groups*, *at* http://www.vhdl.org (last visited Apr. 15, 2004). Verilog became IEEE Standard 1364 in 1995. *Id.*

[140] Alternatively, the reader may identify and obtain an existing machine that performs the functions described by the explanation.

that passes between the explanation's authorship and the final construction of the automobile.

These are all the consequence of the fact, that prior to computers, only humans could understand natural languages (such as English) and respond to instructions expressed in natural languages, with the exception of some domesticated animals. Machines could not.

Software speech, in contrast, may be executed automatically by a computer.[141] It is possible to cause the instructions embodied in software speech, in the form of source code, to be carried out automatically by a computer, without a human intermediary and without the intermediate exercise of any discretion, by providing the source code directly to the computer.[142] Although one continues to hear the lament that we have

[141] Providing a rigorous definition of automation is beyond the scope of this article. At least two plausible meanings, however, are sufficient for purposes of the present discussion. First, an action may be "automatic" if its execution does not involve the performance of substantial physical acts by a human being. This sense of the term "automatic" focuses on the lack of all but the most trivial *physical* involvement by a human being in performance of the action, and is reflected in certain dictionary definitions of "automatic" (e.g., "done or produced as if by machine" and "having a self-acting or self-regulating mechanism") and "automaton" ("a machine or control mechanism designed to follow automatically a predetermined sequence of operations or respond to encoded instructions"). MERRIAM WEBSTER'S COLLEGIATE DICTIONARY 78 (10th ed. 1993). A dishwasher, for example, washes dishes "automatically" in the sense that it performs the necessary sequence of actions without the physical assistance of a human being, other than the physically trivial acts of selecting a setting and pressing the power button. Second, an action may be "automatic" if it is performed without all but the most trivial *mental* involvement by a human being, i.e., if it is performed without the substantial exercise of conscious human thought or discretion. In this second sense of the term "automatic," an action may be performed automatically even if physically performed substantially or entirely by a human so long as the action is performed unconsciously or without the need to exercise substantial discretion. Workers on an industrial assembly line perform their actions "automatically" in this sense, despite the significant amount of physical labor they perform. This meaning of the term "automatic" is also reflected in conventional dictionary definitions (e.g., "largely or wholly involuntary" and "acting or done spontaneously or unconsciously"). *Id.* In the context of software, an executable computer program residing in the memory of a computer may be executed, for example, in response to a user double-clicking on an icon representing the program. Such a program is executable "automatically" in both senses just described; execution of the software requires the human user to engage in neither substantial *physical* activity nor substantial *mental* activity. Executing the program is akin to pressing the "power" button on a dishwasher. Interesting questions, although beyond the scope of this paper, are raised by various meanings of the term "automatic," including whether computer source code written on paper qualifies as an "executable software program" if it may be executed "automatically" by a computer in either or both of the two senses described above. Also beyond the scope of this paper is the question of whether all tangibly-embodied instructions that are executable by a computer qualify as software, or whether such instructions must be stored in the memory of a computer to qualify as software.

[142] Burk, *supra* note 36, at 118–19:

> This attempt to equate functional texts with source code [objecting to injunctions against dissemination of source code on the same grounds as objecting to injunctions against dissemination of non-software speech] is fundamentally flawed because it conflates two different meanings of the word "instruction." Instructions to a computer are not the same as instructions to a human. Unlike recipes or "how to" manuals, source code instructions do not instruct a human how to carry out a process. [Humans typically do not follow source code "instructions" to arrange voltages in computational registers.] Moreover, the human recipient of written instructions may choose not to execute them. The source code may be read by a human, who may understand what processes it entails, but the code remains abstracted machinery – it will be executed by the machine automatically and involuntarily, after being automatically and involuntarily compiled into the proper format. In short, instructions to a human merely describe

lived through the year 2001 without seeing the advent of truly voice-controlled computers *à la* HAL 9000,[143] computers have been *language*-controlled ever since they first became programmable, despite the fact that it typically is necessary to use special computer programming languages, rather than natural human languages, to control computers. Computers are, in effect, the first language-controlled machines. As a result, source code is speech with a kind of power not possessed by any previous kind of speech.

It is worth noting that the assertion that source code is "functional" has engendered much controversy. Some have denied that software is "functional" at all.[144] Others have argued that source code is only "functional" in the same sense as other works which are protected by the First Amendment, such as choreography, musical scores, and architectural blueprints, and that the "functional" nature of source code should therefore be irrelevant to its treatment by the First Amendment.[145]

The analysis above should make clear that source code differs from these other kinds of works in a critical way. Choreography, musical scores, and architectural blueprints all require human discretion and physical effort to execute, unlike software, which may be executed directly and automatically by a machine.[146] The assertion, therefore, that "[t]o allow the government to regulate software based only on this [functional] characteristic means that the government may control many works covered by the First Amendment"[147] is false.

Some who argue that source code is not "functional" appear to conflate two different meanings of the term "functional." The first meaning, used herein, is that an object is "functional" if it can *directly act upon physical entities to achieve a useful result.* An object acts "directly"

> how to perform a task whereas software instructions actually are a part of the machine that executes the task.

Id. See David S. Touretzky, *Source vs. Object Code: A False Dichotomy* (July 12, 2000) (unpublished essay, Carnegie Mellon University), *available at* http://www-2.cs.cmu.edu/~dst/DeCSS/object-code.txt. "All computer code is executable. In some instances it may be advantageous to transform the code into another form first, but transformation is by no means mandatory. An interpreter can be employed instead. Interpreters are in common use in computer systems." *Id.*

[143] ARTHUR C. CLARKE, 2001: A SPACE ODYSSEY 95–97 (1968).

[144] *See supra* note 14.

[145] *See, e.g.,* Tien, *supra* note 6, at 692 (arguing that the fact that source code is executable does not distinguish it from other words protected by the First Amendment, because many "works of expression . . . provide ways of achieving a certain result because they are intended to be 'executed.' Choreography, musical scores, and stage direction are just a few examples of such works."). Andy Oram asserts:

> A distinction between the "expressive" and "functional" aspects of speech winds its way through numerous court cases, including the ones on cryptography mentioned earlier. But computer code is not the only kind of speech that is functional; recipes, directions from an appliance manufacturer, and many other forms of speech also have functional aspects.

Andy Oram, *Copy Controls and Circumvention: Don't Get Around Much Any More* (Jan. 17, 2002), *at* http://www.oreillynet.com/pub/wlg/1068.

[146] *See* Burk, *supra* note 36, at 119.

[147] Tien, *supra* note 6, at 692.

if it acts without substantial intermediating human agency. Although it may be difficult to determine whether a particular act of human intervention is "substantial" in a particular case, there are enough clear cases to make the definition useful. Consider some clear cases of functional objects such as hammers, guns, and automobiles. Although a hammer requires the input of energy by a human to perform its function of pounding nails, the hammer performs its function directly on the nail upon the input of energy by a human. All machines require a human to provide some initial and/or guiding energy to enable the machine to perform its function. We still call such machines "functional" despite the requirement that humans initiate or control some of the machine's actions. Consider, for example, the act of driving a car by applying appropriate forces to the car's steering wheel and pedals. The car automatically performs appropriate functions in response to application of such forces. This is what makes the car functional. A written recipe, in contrast, does not bake a cake. No amount of input energy may be applied to a written recipe to cause it to bake a cake automatically (i.e., without the intermediation of human agency). Rather, the recipe is, in effect, an input to a *human* (the cook), who uses the information provided in the recipe to perform the functions described in the recipe. In other words, human agency is required to perform the functions described in the recipe. In the case of the automobile, the chain of causation is from human to car to function, while in the case of the recipe, the chain of causation is from recipe to human to function.

Confusion arises in this area, in part, from the fact that the term "functional" also has another meaning. A particular written work is sometimes called "functional" if it is written using functional language, i.e., if it describes functions to be performed. Recipes, for example, are typically written in functional language. In particular, recipes are written as a set of instructions expressed in the imperative tense—"do A, then do B, then do C," and so on. Software source code also typically describes a set of functions to be performed in the imperative tense. In the case of a recipe, the intended audience is a human chef, while in the case of software the intended audience is a computer. It is true that both a recipe and source code are "functional" in this second sense. A recipe, however, cannot perform the functions it describes directly, i.e., without the substantial intervention of human agency. A recipe, therefore, is not functional in the first sense described above, which is the sense that is relevant to the analysis herein.

Software, therefore, is functional in a way that differs qualitatively from other tangible embodiments of expression. The unique causal power of software speech is no accident. Rather, it is part of the very nature of software and is the conscious result of the design of the modern general-purpose digital computer architecture. The founders of modern computing recognized that causing a machine to perform a new function almost always required engineers to design and painstakingly build the physical structure of a new machine or modify the physical structure of an existing machine. This was a tedious, time-consuming, error-prone, and costly process. Computers were invented to simplify the process of causing machines to perform new functions.[148]

The particular solution that was chosen to solve the problem of machine design was a machine called the general-purpose computer or "universal machine," which could be made to perform nearly any function merely by providing it with instructions—in the form of software—to do so, thereby obviating the need to design and build a new physical machine each time a new function was desired. The goal of implementing such a machine has been achieved, perhaps beyond all original expectations.[149] Although it would have been difficult to believe

[148] The ENIAC, for example, was "[a]n enormous machine, occupying a large room, and programmed by connecting cables to a plugboard rather like an old-fashioned telephone switchboard." MARTIN DAVIS, ENGINES OF LOGIC: MATHEMATICIANS AND THE ORIGIN OF THE COMPUTER 181 (2001). *See also* GEORGES IFRAH, THE UNIVERSAL HISTORY OF COMPUTING: FROM THE ABACUS TO THE QUANTUM COMPUTER 281 (2001) ("[E]ach time a new sequence of events was to be carried out [by ENIAC], the programme had to be changed by making radical modifications to the highly complex system of connection panels ("patch boards"). The machine was not usable until this lengthy and tedious manual work was completed."). Jon Agar states:

> The ENIAC . . . could also be 'reprogrammed,' switching from calculating trajectories to the motion of a shockwave. But this reprogramming meant extensive rewiring: two days spent completely rearranging a nest of plugboard wires. It was not the same machine doing different calculations, but a subtly different machine for each job. . . . [T]he frustration of rebuilding the machine each time the mathematical problem was changed forced the ENIAC team to devise the crucial idea of this book: the concept of the stored program.

JOHN AGAR, TURING AND THE UNIVERSAL MACHINE 60–61 (2001).

The quest to eliminate the tedium of electromechanical design is related to the general quest of scientists to eliminate the tedium involved in performing the calculations that are necessary for all scientific work. Ifrah argues:

> But if the world is ruled by numbers, the world is not amused thereby. Calculation is slow, difficult and above all tedious. It is tedious because it is repetitive, and this has given rise to all sorts of hindrances to the advancement of science and knowledge. Luigi Menabrea's words on this theme are heavy with meaning: "What quantities of previous observations have been of no use to the advancement of science and technology, for the reason that there were not the resources needed to calculate results from them! How the prospects of long and arid calculation have demoralized great thinkers, who seek only time to meditate but instead see themselves swamped by the sheer mass of arithmetic to be done by an inadequate system! And yet, it is the path of laborious analysis that leads to truth, but they cannot follow that path without employing the guide of number, for without number there is no way to lift the veil which covers the mysteries of nature."

IFRAH, *supra* at 101.

[149] *See* A.M. Turing, *On Computable Numbers With an Application to The Entscheidungsproblem,* PROC. LONDON MATH. SOC., 42 (1936), 230–65, correction in PROC. LONDON MATH. SOC., 43 (1937), 544–46 (describing for the first time what is now referred to as a "universal machine"). For general discussion of the paper and the work behind it, see AGAR, *supra* note 148, at 139–76. The "universal machine" is also referred to as the "universal Turing machine," which is capable of mimicking any of a near-infinite number of special-purpose machines referred to as "Turing machines."

According to Turing, the universal machine's logical architecture was comprised essentially of a memory for storing instructions (software), and a processing unit for retrieving instructions from the memory, carrying out the instructions, and storing the results of the instructions back into the memory. *Id.* The architecture may also include an input unit for receiving data from a human user or an external device, and an output unit for providing data to a human user or external device. *See, e.g.,* DAVIS, *supra* note 148, at 151. Ifrah outlines:

> Third definition: A computer is an artificial automaton comprising a facility for input and output, a memory, a processor capable of effecting all sorts of transformations on data expressed in the

one hundred years ago that a single machine could balance checkbooks, play games, perform mathematical calculations, display artwork and movies, and transmit messages across the world, a single modern computer can perform all of these different functions and more.[150]

In general, it is easier, quicker, and less expensive to design, implement, and execute a software program than it is to design, build, and operate hardware that performs the same functions. The power to make machines perform actions using words is what attracts many people to programming, and is one of the primary reasons for the rapid rate of progress in the field of software and the widespread adoption of software generally. Software has reduced the amount of mental and physical effort needed to cause machines to perform new functions and, more generally, has reduced the length of the causal chain connecting idea to instruction to action.

More recently, the advent and widespread proliferation of the Internet and networking technologies has made it possible to transmit, copy, and store information more rapidly, easily, and inexpensively than ever before.[151] Because software may be embodied in a tangible form (digital signals) that is capable of being transmitted over the Internet, the Internet has further facilitated the process of causing machines to perform new functions. A software program, such as malicious virus software or a benign operating system patch, typically may be distributed to a large number of geographically-dispersed computers rapidly, easily, and inexpensively, and then executed instantly.

The particular abilities of software to control computers and to be transmitted, copied, and stored at a very low cost combine to give software speech causal powers exceeding those of most comparable conventional speech, all other things being equal.[152]

> form of character-chains (material representations of encoded data) and which, within the limits of its physical capacities, permits the execution of all types of symbolic calculations (and thus the solving of all problems where the solution may be expressed in the form of an algorithm), governed by a control unit instructed by programmes input into the memory (and thus handling the commands to be enacted in the same way as the data to process).

IFRAH, *supra* note 148, at 313.

[150] Even Howard Aiken, one of the founders of modern computing, expressed his surprise at the ability of a computer to mimic a wide variety of machines when he said: "If it should turn out the basic logics of a machine designed for the numerical solution of differential equations coincide with the logics of a machine intended to make bills for a department store, I would regard this as the most amazing coincidence I have ever encountered." PAUL E. CERUZZI, RECKONERS: THE PREHISTORY OF THE DIGITAL COMPUTER, FROM RELAYS TO THE STORED PROGRAM CONCEPT 1935–1945, 43 (1983) (quoting Howard Aiken, *The Future of Automatic Computing Machinery*, ELEKTRONISCHE RECHERANLAGE UND INFORMATIONSVERARBEITUNG 33, (1956)).

[151] *See, e.g.,* Pamela Samuelson, *Digital Media and the Law*, 34 COMM. OF THE ACM 10, 23–28 (1991) (describing six distinctive qualities of the digital medium that "may be the reason why so many challenging legal issues have arisen over the past few years": (1) ease of replication; (2) ease of transmission and multiple use; (3) plasticity of digital media; (4) equivalence of works in digital form; (5) compactness of works in digital form; and (6) nonlinearity).

[152] I explain what I mean by "comparable" and "all other things being equal" in more detail in Part IV.F, *infra*.

F. Software Speech and Conventional Speech Compared

In the previous section, I argued that software speech generally has greater causal powers than conventional speech; that is, software speech is more "powerful" than conventional speech. One way to appreciate the manner in which software speech is more powerful than conventional speech, at least in some circumstances, is by considering paired analogous cases of software speech and conventional speech. Consider the following hypothetical, which is intended to be analogous to the virus hypotheticals above.

A hardware engineer devises an innovative technique for erasing information from computer hard drives from a relatively long distance (a few feet) using relatively weak magnets (small enough to be hidden in a pocket) having a particular novel shape which gives them their data-erasing properties. The hardware engineer writes detailed instructions explaining how to forge such magnets and how to use them to erase computer data, and mails the instructions by postal mail to one thousand recipients selected at random. The engineer composes the instructions as a chain letter, instructing the recipients to forward the instructions to ten more recipients of their own choosing. Without going through each permutation, one may imagine different cases in which the hardware engineer has each of the different levels of intent and mixtures of motives described above with respect to the virus author. In one case, for example, the hardware engineer encourages people to construct the novel magnets and to walk about their workplaces with the magnets in their pockets to enhance their libidos, thereby unwittingly enlisting an army of data destroyers.

Comparing this family of hypotheticals to the virus hypotheticals, I assert that the causal chain connecting the virus author to the harm caused by the virus is likely to be much shorter and more direct than the causal chain connecting the hardware engineer to any harm caused by someone who follows his instructions.[153] To follow the hardware

[153] One might wonder whether the actions of someone who redistributes the virus or carries out the magnet instructions qualify as an intervening force of the kind that would qualify as a superseding cause of the harm caused by the virus/magnet, thereby absolving the virus/magnet author of liability. *See* RESTATEMENT (THIRD) OF TORTS § 34 (Tentative Draft No. 3, 2003) ("An actor is not subject to liability for harm, for which a force of nature or an independent act is also a factual cause of the harm, if the harm is different from the harms whose risk made the actor's conduct tortious."); RESTATEMENT (SECOND) OF TORTS § 441(1) (1965) ("An intervening force is one which actively operates in producing harm to another after the actor's negligent act or omission has been committed."); *id.* § 442 (listing six considerations that are important in determining whether an intervening force is a superseding cause); *id.* § 442A ("Where the negligent conduct of the actor creates or increases the foreseeable risk of harm through the intervention of another force, and is a substantial factor in causing the harm, such intervention is not a superseding cause."); *id.* § 435(1) ("If the actor's conduct is a substantial factor in bringing about harm to another, the fact that the actor neither foresaw nor should have foreseen the extent of the harm or the manner in which it occurred does not prevent him from being liable."); *id.* § 440 ("A superseding cause is an act of a third person or other force which by its intervention prevents the actor from being liable for harm to another which his antecedent negligence is a substantial factor in bringing about.").

engineer's instructions, one would likely need to exercise a fairly significant amount of independent discretion and judgment, and one would need to engage in planning, materials gathering, and other physical activity, all spread out over an appreciable amount of time. In contrast, following the virus author's instructions requires little more than the intent to do so. It should not be surprising that the source code's instructions may be carried out at a lower cost, in the broadest sense of that term, than the hardware engineer's instructions for two reasons: (1) in general, software may be designed and implemented at less expense than hardware that performs the same functions; and (2) in general, the cost of distributing software over the Internet is lower than the cost of distributing objects composed of matter.[154]

In one sense, the impact of software is no different from the impact of virtually any new and useful technology. The introduction of any new device typically lowers the cost of performing the device's intended function in comparison to pre-existing devices for performing the same function. If a new device did not have this effect, it would likely be of little use. The domestication of the horse lowered the cost[155] of transportation, as did the subsequent introduction of the automobile and the airplane. Put simply, it is easier to travel cross-country in an airplane than on a horse.

Lowering the cost of performing a function makes it possible to shorten the causal chain linking the intent to perform a particular action to its actual performance. Consider, for example, the effect of the

I do not address this question here because there is no apparent reason to believe that the answer to this question will be different in software speech cases than it is in conventional speech cases, and the focus of the present analysis is on potential differences in outcome between software speech cases and conventional speech cases. Indeed, to the extent that it is more foreseeable that software speech in the form of software, such as a virus, will be maliciously redistributed by a human intermediary than it is that English-language instructions will be maliciously carried out by a human intermediary, the case for superseding cause may be weaker (and the case for liability therefore stronger) in the case of software speakers than in the case of conventional speakers. *See* RESTATEMENT (THIRD) OF TORTS § 34 cmt. d, illus. 1 ("Even when the risk posed by the intervening act is not *the* reason for finding the actor's conduct to be tortious, many courts have accepted the proposition that, so long as the intervening act was foreseeable to the actor, the intervening act is not a superseding cause."); RESTATEMENT (SECOND) OF TORTS § 435(2) ("The actor's conduct may be held not to be a legal cause of harm to another where after the event and looking back from the harm to the actor's negligent conduct, it appears to the court highly extraordinary that it should have brought about the harm."). In either software speech or conventional speech cases, if the negligence of the speaker creates or increases the foreseeable risk of harm through the intervention of another force, and is a substantial factor in causing the harm, such intervention is not a superseding cause, even if such intervention is the act of a human being. *Id.* § 442A. That is, "whoever acts negligently is answerable for all the consequences that may ensue in the ordinary course of events, even though such consequences are immediately and directly brought about by an intervening cause, if such intervening cause was set in motion by the original wrongdoer." 57A AM. JUR. 2D *Negligence* § 618 (1989).

[154] *See* Halpern, *supra* note 6, at 165–66 ("[T]he primary difference between source code and an instructional textbook is the degree of ease by which the reader can transform acquired knowledge [from the source code] into the electromagnetic tangible known as object code.").

[155] I use the term "cost" in its most general sense to include factors such as monetary expense and transaction costs. In the context of transportation, cost may be measured, for example, in cost per mile traveled.

introduction of guns. A gun makes it easier to kill, given the intent to kill; in other words, guns make it possible to lower the cost of killing. The power of guns allows someone, who has the intent to kill and who possesses a gun, to kill more easily than someone with the same intent but in possession of a less powerful weapon.[156] It is in this sense that I assert that a gun has greater causal powers than less powerful weapons. Therefore, when a person with the intent to kill uses a gun to do so, the causal chain connecting the killer to the victim's death is shorter and more direct than in comparable cases involving less powerful weapons, all other things being equal.

Although both software and conventional technological devices make it easier to perform particular functions, what distinguishes software from conventional technological devices for purposes of the present discussion is that software uses *language* embodied in a tangible form as the means for causing a desired function to be performed. What distinguishes software from other forms of expression, such as oral and written instructions expressed in natural languages, is that the act of expressing ideas using software can do more than merely express ideas. In particular, the act of expressing ideas in the form of software code can cause the expressed functions to be performed automatically. Whether software is a machine "constructed in the medium of text,"[157] or whether it is speech having properties of a machine, is irrelevant. What is critical is that source code can both be expressive, in the same sense as speech expressed in natural human languages, and have causal powers that are qualitatively different from other forms of speech.[158]

G. Conclusion: Software Speech is Less Protected Than Conventional Speech

Based on the observations above, I argue below that software speech is generally subject to a lesser degree of First Amendment protection than conventional speech having identical expressive content. This is true even if the same, stringent, First Amendment standards are correctly applied to both kinds of speech.[159] This reduced protection afforded to software speech is not necessarily the result of any legislative intent to suppress speech or judicial ignorance of the expressive

[156] It is critical for purposes of this discussion to limit this assertion to cases in which the intent to kill has already been formed. If the intent to kill has not already been formed, the possession of a gun may make it either more or less likely that the gun-possessor will form the intent to kill. Knowledge of the gun's power may, for example, deter the possessor from using the gun at all.

[157] Samuelson, *supra* note 136, at 2327 (arguing that computer programs are "machines that happen to have been constructed in the medium of text" and therefore differ from other kinds of machines and other kinds of textual works for purposes of intellectual property protection).

[158] See *supra* Part I.C.1 for discussion of the fact that a single software program can be both highly expressive and highly functional.

[159] This statement is intended as an observation about the application of current law to software speech and conventional speech, not as a recommendation for how the law should apply to software speech.

capabilities of source code.[160] Rather, software speech may obtain less protection than comparable conventional speech in particular classes of cases even when the legislature has no intent to suppress speech and the courts properly apply highly protective First Amendment standards to software speech. The fundamental reason for the difference in protection is not censorious intent on the part of legislators or judges, but rather a fundamental tension between competing values that are deeply rooted in the libertarian tradition itself.[161] These are being brought to the surface in increasingly vexing ways by the advent of software.[162]

The relative difference in protection afforded to software speech and conventional speech should appear most clearly at the margins, i.e., in cases in which the elements of intent and proximate cause border most closely on being satisfied. In particular, I assert that if a plaintiff falls just short of proving the elements of intent and proximate cause against a defendant in a case in which the defendant's protected conventional speech indirectly brings harm to a third party (the plaintiff) through an intermediary, the same plaintiff could justifiably succeed in proving the elements of intent and proximate cause if the case instead involved software speech having the same expressive content, assuming everything else is equal. This leads to the perhaps surprising result that in two cases involving speech having the same protected expressive content, one defendant may be found liable and the other not liable depending on whether their speech is expressed in, and/or carried out using, software.

[160] Of course, in particular cases there may be such censorious intent.

[161] These competing values are reflected in the harm principle, according to which individual liberty may be restrained to prevent harm to others. Regarding the classic statement of the harm principle, John Stuart Mill wrote:

> The object of this Essay is to assert one very simple principle, as entitled to govern absolutely the dealings of society with the individual in the way of compulsion and control, whether the means used be physical force in the form of legal penalties or the moral coercion of public opinion. That principle is that the sole end for which mankind are warranted, individually or collectively, in interfering with the liberty of action of any of their number is self-protection. That the only purpose for which power can be rightfully exercised over any member of a civilized community, against his will, is to prevent harm to others. His own good, either physical or moral, is not a sufficient warrant. He cannot rightfully be compelled to do or forbear because it will be better for him to do so, because it will make him happier, because, in the opinions of others, to do so would be wise or even right. These are good reasons for remonstrating with him, or reasoning with him, or persuading him, or entreating him, but not for compelling him, or visiting him with any evil in case he do otherwise. To justify that, the conduct from which it is desired to deter him must be calculated to produce evil to someone else. The only part of the conduct of anyone for which he is amenable to society is that which concerns others. In the part which merely concerns himself, his independence is, of right, absolute. Over himself, over his own body and mind, the individual is sovereign.

JOHN STUART MILL, ON LIBERTY 16 (Prometheus Books 1986) (1859).

[162] One commentator has noted that mass media speech is potentially susceptible to greater liability in common law negligence actions for similar reasons. Andrew Sims notes:

> [T]he broad dissemination of ideas and information which make the mass media valuable to society from a First Amendment standpoint make them potentially more vulnerable as defendants in common law negligence actions, both in terms of the magnitude of the risk presented and the correlative duty imposed, and in terms of the potential plaintiff class. The more widely-disseminated the speech, the greater the potential tort liability exposure.

Sims, *supra* note 93, at 274.

Consider, for example, the virus software hypothetical in which the virus author naively releases the virus software, intending that it solely be used for educational purposes. One might imagine a negligence claim being sustained against the virus author in such a case and the same negligence action failing against the hardware engineer who naively mails his magnet instructions. Even if the proximate cause element of the claim incorporates the *Brandenburg* "imminence" requirement,[163] the speed and ease with which instructions in software speech may be distributed and carried out may justify a finding of imminence in the case of the virus author but not in the case of the hardware engineer. In other words, the case for proximate cause is likely to be stronger in the case of software speech than in the case of conventional speech because the causal chain linking the software speaker to the resulting harm is likely to be shorter and stronger than the causal chain linking the conventional speaker to the harm caused by his speech.[164]

Furthermore, the context in which speech is expressed is relevant to determining the intent of the speaker.[165] In particular, a factfinder may take into account a speaker's knowledge of the foreseeable consequences

[163] *See supra* note 79.

[164] For a similar argument about the causal chain in cases of violence caused by speech of the mass media, see Sims, *supra* note 93, at 260. Although I discuss proximate cause in terms of the strengths of causal chains, the analysis provided herein does not rely on this or any particular conception of proximate cause. The meaning of "proximate cause" has been, and continues to be, the subject of much confusion:

> There is perhaps nothing in the entire field of law which has called forth more disagreement, or upon which the opinions are in such a welter of confusion [than the meaning of "proximate cause"]. Nor, despite manifold attempts which have been made to clarify the subject, is there yet any general agreement as to the best approach.

KEETON ET AL., *supra* note 72, § 41.

Furthermore, the question of proximate cause (also referred to as "legal cause") is often confused with the question of factual cause (also referred to as "actual cause"). 37 AM. JUR. 2D, *Proof of Facts* § 3, at 533 (1984) ("Typically the various doctrines of proximate causation have blended and obscured actual causation and such legal policies, serving as a limitation which the courts have placed on the actor's responsibility for the consequences of his conduct.").

[165] *See generally* Planned Parenthood of Columbia/Willamette, Inc. v. Am. Coalition of Life Activists, 290 F.3d 1058 (9th Cir. 2002). Tort law recognizes that the context in which an actor acts may be evidence of the actor's actual intent:

> The factfinder [in a tort case] need not credit the actor's assertion that the actor did not intend the result in question. One of the common lines of argument against crediting the actor's assertion is (1) that, given the circumstances disclosed in the evidence, a reasonable person in the actor's position would have known that the consequence in question was substantially certain to follow the act, (2) that the evidence shows that the actor was even brighter and shrewder than most others, and (3) that the inference is therefore compelling that the actor knew even though testifying otherwise. If the factfinder credits inference (1) but not inferences (2) and (3), the finding is negligence. But if the factfinder credits all three inferences, the finding is intent to produce the consequence in question. Expressed another way the point is this: Since intent is a state of mind, it is plainly incorrect for a court to instruct a jury that an actor is presumed to intend the natural and probable consequences of the actor's conduct; but it is correct to tell the jury that, relying on circumstantial evidence, they may infer that the actor's state of mind was the same as a reasonable person's state of mind would have been. Thus, when the driver who whips up horses with a loud yell while passing a neighbor's team denies intent to cause a runaway, the factfinder may discredit the driver's testimony; and the defendant on a bicycle who rides down a person in full view on a sidewalk where there is ample room to pass may learn that the factfinder (judge or jury) is unwilling to credit the statement, "I didn't mean to do it."

KEETON ET AL., *supra* note 72, § 41, at 263.

of his speech to ascertain the speaker's intent.[166] Even if *Sullivan*'s recklessness intent requirement is incorporated into the negligence claim, naively distributing a virus onto the Internet may be taken as evidence of recklessness rather than mere negligence if the virus author is, or should be, aware of the ability of viruses to be propagated quickly and easily over the Internet.[167] In such a case, the speaker may be charged with a higher degree of intent, perhaps justifiably, than in a case involving conventional speech having the same expressive content.[168]

Once again, my intent is not to argue for particular outcomes in particular hypothetical cases, but rather to demonstrate more generally that it is possible to identify pairs of corresponding cases involving speech with the same expressive content but different legal outcomes, depending on whether the speech is embodied in software.

[166] In the tort context, for example, a defendant may be charged with intent if the circumstances warrant a finding that there was a substantial certainty that the defendant's actions would result in a known danger. If the occurrence of the known danger is merely foreseeable rather than a substantial certainty, at most a finding of negligence is justified. *Id.* § 8, at 36.

The likely consequences of speech are not, however, the only factors relevant to intent. There may be circumstances when a speaker has no choice or is otherwise justified in speaking despite the likely consequences of her speech. For example, speakers in public places are often greeted with hostility or heckling by their audiences. It may be foreseeable to a speaker in such a circumstance that to continue speaking will cause a breach of the peace. To suppress the speaker's speech in such a circumstance, however, would be to sanction the "heckler's veto." *See generally* David G. Barnum, *Freedom of Assembly and the Hostile Audience in Anglo-American Law,* 29 AM. J. COMP. L. 59, 74–75 (1981) (arguing that "the police must be prepared to intervene in some situations by arresting demonstrators and in others by controlling or arresting the members of the hostile audience.").

[167] *See infra* note 172.

[168] Other claim elements may also be easier to prove in software speech cases than in corresponding conventional speech cases. Consider, for example, the duty element of a negligence claim. Some jurisdictions use a "foreseeability" test to determine whether the defendant owed the plaintiff a duty of care. 37 AM. JUR. 2D *Proof of Facts* § 3, at 534 (1984). According to such a test, the defendant owes the plaintiff a duty of care only if the harm caused to the negligence plaintiff was foreseeable. *Id.* There is a strong case to be made that the degree of foreseeability in each virus hypothetical is higher than the degree of foreseeability in each corresponding magnet hypothetical, with respect to the resulting harm. Consider, for example, the purely benign virus author who distributes the virus without first warning the operating system manufacturer of the virus. It is more foreseeable that such action will cause harm than it is that the corresponding magnet author's action will cause harm. One might also compare the foreseeability of harm in the virus hypotheticals to the foreseeability of harm in various cases involving harmful actions allegedly inspired by violent television shows, movies, or songs. *See generally* Richard M. Goehler & Jill Meyer Vollman, *Expansion of Tort Law at the Expense of the First Amendment: Has the* Jones *Court Gone Too Far? Stay Tuned to Find Out*, 27 N. KY. L. REV. 112 (2000). In most circumstances, it is not foreseeable that the distribution of a violent film or song will cause someone to mimic the violence portrayed therein or to otherwise engage in physical violence. *See generally* McCollum v. CBS, Inc., 249 Cal. Rptr. 187 (Cal. Ct. App. 1988). It is significantly more foreseeable that virus source code made freely available on the Internet will be redistributed maliciously by at least one person to whom it is made available. As mentioned earlier in this Article, some jurisdictions use foreseeability as a test for the existence of proximate cause. 37 AM. JUR. 2D, *Proof of Facts* § 3, at 534 (1984). Such use of foreseeability, however, "has been criticized on the ground that, *inter alia*, it is unwarrantably duplicative and conceptually confusing since foreseeability in the negligence field is more appropriately allocable to the sphere of definition of duty than to proximate causation." *Id.* at n.33.

V. A SIMPLIFIED MODEL OF INTENT AND CAUSATION

Arguments for strong First Amendment protection for source code have tended to assume or expressly assert that the kind and degree of First Amendment protection afforded to speech should not vary based merely on the form that the speech takes.[169] The contrary conclusion drawn above[170]—that the degree of protection afforded to speech may vary depending on whether such speech is embodied in software—will now be further explained using a simplified model of liability, in which the variable I represents intent and the variable C represents proximate causation. More specifically, the variable I represents the "strength" or the "degree" of the defendant's intent, wherein higher values of I represent stronger intent. The variable C represents the strength of the causal chain linking the defendant's actions to the alleged harm, wherein higher values of C represent a stronger causal chain. Although variables and mathematical formulae are used in the analysis that follows for clarity of explanation, such an analysis can only be applied in extremely rough approximation in actual cases. The use of mathematical language in the following analysis is intended only to aid in understanding the relative degrees to which the elements of intent and proximate cause are satisfied in particular cases, even though it may not be possible in reality to assign quantitative values to such elements.

Referring to Figure 1, a diagram is shown which illustrates a continuum of "causal chain strength," represented by the variable C.[171] The value of C ranges from a minimum value of zero, indicating no causal connection at all between the defendant's actions and the alleged

[169] One commentator, for example, has "argue[d] that the content of software is identical regardless of its form—be it on paper or in a computer—and thus deserving of coverage," and has criticized drawing any First Amendment distinction between expressions of algorithms based merely on whether such expressions take the form of English-language descriptions or computer source code. Tien, *supra* note 6, at 655. Such arguments imply that software speech and conventional speech should receive the same degree of protection if they have the same expressive content. The same commentator asserts that "it would sound strange to say that the First Amendment does not cover some speech simply because a computer could execute it," thereby expressing the belief that speech which is protected in one form should not lose its protection if it is expressed in the form of executable source code. *Id.* at 693; *see also Bernstein III*, 974 F. Supp. 1288, 1307 (N.D. Cal. 1997) ("[T]he dramatically different treatment of the same materials depending on the medium by which they are conveyed is not only irrational, it may be impermissible under traditional First Amendment analysis.").

> Like choice of language or functionality of speech, choice of medium cannot be allowed to drive First Amendment doctrine. Inscribing information onto a floppy disk or other magnetic medium does nothing to alter the information itself, and should do nothing to alter the protections offered by the First Amendment. . . . To distinguish between information stored on a floppy disk and information stored in a book is to assert that information on floppy disk is qualitatively different than that on paper. The Court should reject such an assertion. Such reasoning would lead to the conclusion that while a printed newspaper is entitled to First Amendment protection, an electronically published newspaper is not.

Brief of Amicus Curiae Garrett Epps, at Part II.D, *Bernstein IV* (No. 97-16686), *available at* http://www.eff.org/bernstein/Legal/971110_lawprofs.amicus (last visited Apr. 15, 2004).

[170] *See supra* Part IV.G.

[171] The present discussion applies equally to the defendant's intent, *I*, which could be represented by a similar diagram.

harm, to a maximum value of C_{MAX}, representing the strongest possible causal connection between the defendant's actions and the alleged harm.

Assume that liability for a particular cause of action requires that certain minimum thresholds of intent and causation, represented respectively by the variables I_{thresh} and C_{thresh}, be satisfied. Let the actual intent and proximate cause, evidenced by the facts of a particular case, be represented by the variables I_D and C_D, respectively, where "D" stands for "defendant." The defendant thus may be held liable only if the defendant's intent I_D satisfies the minimum intent threshold I_{thresh} (i.e., if $I_D \bullet I_{thresh}$) and the causal connection C_D linking the defendant's actions to the alleged harm satisfies the minimum causation threshold (i.e., if $C_D \bullet C_{thresh}$). Referring again to Figure 1, the defendant's actions satisfy the proximate cause requirement when the value of C_D is within the regions labeled (3), (4), and (5) (all of which are above the boundary line established by the threshold value C_{thresh}), while the defendant's actions do not satisfy the proximate cause requirement when the value of C_D is within the regions labeled (1) and (2) (both of which are below the boundary line established by the threshold value C_{thresh}).

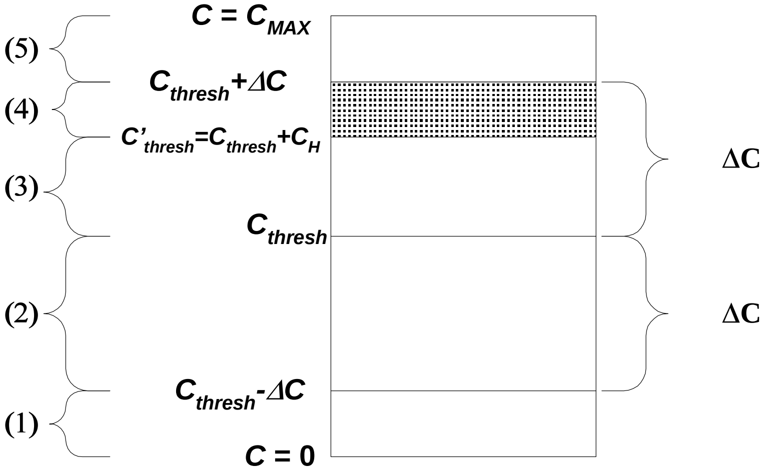

Figure 1

Now imagine two cases, a first case involving conventional speech and a second case involving software speech both having the same expressive content and all else being equal. Let I_0 be the intent of the conventional speaker, and let C_0 be the strength of the causal chain linking the conventional speaker's speech to the alleged harm in the first case. Similarly, let I_1 be the intent of the software speaker (i.e., the programmer), and let C_1 be the strength of the causal chain linking the programmer's speech (embodied in source code) to the alleged harm in the second case.

Now let I_L (where "L" stands for "liability") be the difference between the defendant's intent I_D and the minimum required intent I_{thresh}. In other words, $I_L = I_D - I_{thresh}$. Similarly, let C_L be the difference between the strength C_D of the causal chain linking the defendant D's actions to the harm in question and the minimum required causal chain strength C_{thresh}. In other words, $C_L = C_D - C_{thresh}$. If the defendant's intent I_D satisfies the minimum intent requirement I_{thresh}, then I_L is positive; otherwise, I_L is negative. Similarly, if the causal chain strength C_D satisfies the proximate cause requirement C_{thresh}, then C_L is positive; otherwise, C_L is negative. The defendant D in a particular case may only be held liable if both I_L and C_L are positive and the plaintiff proves the remaining elements of the claim (such as duty and damages, in the case of a negligence claim).

Now let $\bullet I$ be the difference between the intent I_1 of the programmer and the intent I_0 of the conventional speaker. In other words, $\bullet I = I_1 \text{-} I_0$. Similarly, let $\bullet C$ be the difference between the causal chain strength C_1 associated with the programmer and the causal chain strength C_0 associated with the conventional speaker. In other words, $\bullet C = C_1 \text{-} C_0$.

If the analysis above is correct, both $\bullet I$ and $\bullet C$ will tend to be positive as a result of: (1) the causal powers of software (leading to positive values of $\bullet C$); and (2) the defendant programmers' knowledge of the causal powers of software (leading to positive values of $\bullet I$).[172] This is merely another way of stating that in pairs of corresponding conventional speech and software speech cases: (1) the causal link between software speech and its resulting harm is likely to be stronger than the causal link between conventional speech and its resulting harm; and (2) programmers are likely to have a higher degree of intent than conventional speakers in such cases because programmers will tend to know that software speech may be used more easily to perform actions than conventional speech.

This is not to say that $\bullet I$ and $\bullet C$ will *always* be positive, only that they will tend to be positive in typical cases. The present argument does

[172] As described *supra* in notes 165 and 166, the fact-finder may take circumstantial evidence into account to ascertain the intent of the defendant. In cases involving software speech, the fact-finder may find that the programmer-defendant has a higher degree of intent than a corresponding conventional speaker, based on the circumstantial evidence that the programmer-defendant: (1) knows that software may be executed easily once it is obtained; and (2) knows that software may be distributed quickly and easily over the Internet. The fact-finder may not credit, for example, the testimony of a programmer who releases the source code for a novel computer virus on the Internet and claim that he did not intend for the virus to cause harm. In this way, the same features of software and the Internet that may increase the value of C_D (representing the causal connection between the defendant's actions and the alleged harm) may "feed back" into, and thereby result in correspondingly higher values of, I_D (the defendant's intent).

not require that $\bullet I$ and $\bullet C$ always be positive, and it does not imply that courts must find that $\bullet I$ and $\bullet C$ are positive as a matter of law. Rather, the present analysis provides justification for the conclusion that $\bullet I$ and $\bullet C$ will be positive in typical cases based on the application of conventional legal principles to the facts of each case. As a result, programmers will tend to fare worse as defendants than conventional speakers on the elements of intent and proximate cause, even when the expressive content of their speech is identical and the facts of their cases are equivalent in all other respects.

Furthermore, the present argument does not imply that in corresponding cases programmers will always be liable and that conventional speakers will never be liable. Rather, it is possible in any particular pair of corresponding cases for: (1) both the programmer and conventional speaker to be liable; (2) for neither the programmer nor the conventional speaker to be liable; (3) for the programmer, but not the conventional speaker, to be liable; or (4) for the conventional speaker, but not the programmer, to be liable. The present argument does imply, however, that the class of cases in which the programmer satisfies the elements of intent and proximate cause will be relatively large compared to the class of cases in which the conventional speaker satisfies those elements. In particular, the present argument implies that there is an identifiable class of corresponding (paired) cases in which the programmer, but not the conventional speaker, satisfies the elements of intent and proximate cause, and therefore cases in which the programmer, but not the conventional speaker, may be liable.

This class of cases is particularly interesting because it represents a counterexample to the claim that the degree of protection afforded to speech should not vary based merely on the form of the speech.[173] In this class of cases, $C_0 < C_{min}$ (i.e., the causal chain strength associated with the conventional speaker does not satisfy the proximate cause requirement) and $\mathrm{C}_1 \bullet C_{min}$ (i.e., the causal chain strength associated with the programmer-speaker satisfies the proximate cause requirement). Recall that $\bullet C$ is defined as the difference between C_1 and C_0. Therefore, $\bullet C \bullet C_{thresh}$-$C_0$ in this class of cases. In reference to Figure 1, this is evident by observing that when C_0 is below C_{thresh} and C_1 is above C_{thresh}, the difference between C_1 and C_0 ($\bullet C$) must be at least equal to the difference between C_{thresh} and $C_{0.}$

If, as assumed above, all facts of corresponding cases are equivalent except for the fact that the programmer's speech is embodied in software and the conventional speaker's speech is not, it is reasonable to conclude that $\bullet C$ (the difference between the causal chain strengths associated with the programmer and the conventional speaker) is solely attributable to the fact that the programmer's speech is embodied in software rather than in conventional speech. In other words, $\bullet C$ may be interpreted as

[173] *See supra* note 169.

the additional (marginal) causal power of the programmer's speech in comparison to the conventional speaker's speech. Therefore, the implication of the analysis above is that in cases in which the conventional speaker's speech does not satisfy the proximate cause requirement (i.e., in which $C_0 < C_{thresh}$), the programmer's speech will only satisfy the proximate cause requirement if the additional causal power of the programmer's speech (i.e., $\bullet C$) is greater than the extent to which the conventional speaker falls short of satisfying the proximate cause requirement (i.e., C_{thresh}-C_0).

The strength of the causal connection between particular speech and its resulting harm is an empirical question. If the causal power of a programmer's speech is never greater than the causal power of a corresponding conventional speaker's speech, the interesting class of cases described above will be empty, and there will not be any cases in which the programmer-speaker is liable for harm caused by software speech having the same expressive content as conventional speech under the same circumstances. If, however, it is true that $\bullet C$ tends to be positive (i.e., if the causal chain strength C_1 associated with the programmer tends to be greater than the causal chain strength C_0 associated with the conventional speaker), and if it is true that this tendency is not merely a coincidence but rather is a result of the enhanced causal powers of software in comparison to conventional speech, certain important consequences result. For example, in cases in which the conventional speaker falls so far short of satisfying the proximate cause requirement that $C_0 < C_{thresh} - \bullet C$, the programmer will also not satisfy the proximate cause requirement because it follows in such a case that $C_1 < C_{thresh}$. In such a case, therefore, neither the conventional speaker nor the programmer will be liable. This class of cases is represented in Figure 1 by values of C_0 that fall within region (1), and in which C_1 therefore falls within region (2), both of which are below the threshold C_{thresh}.

If we assume that $\bullet C$ is a constant,[174] as the strength C_0 of the conventional speaker's causal chain increases, there will come a point at which the conventional speaker will still not satisfy the proximate cause requirement, but at which the programmer will. That point is at $C_0 =$

[174] To assume that $\bullet C$ is a constant is to assume that the marginal causal power of software speech is the same in all circumstances. It is far from clear whether this assumption is valid. In cases in which the causal chain linking the conventional speaker to the alleged harm is particularly strong, for example, alternatively embodying the speaker's speech in software may have little effect on the strength of the causal chain, thereby representing a small value of $\bullet C$. The same may be true in cases in which the causal chain linking the conventional speaker to the alleged harm is particularly weak. In cases in which the initial causal chain is moderately strong, alternatively embodying the speech in software may provide a significant increase in causal chain strength. This would imply some kind of bell-shaped curve relating causal chain strength to the marginal increase in causal chain strength attributable to the fact that the speech in question is embodied in software. All of this, however, is little more than speculation in the absence of further analysis and possibly empirical study; for purposes of the present argument, it is merely assumed that $\bullet C$ is a constant.

C_{thresh}-$\bullet C$, which represents the lower boundary of the interesting class of cases described above. More generally, the interesting class of cases is represented by values of C_0 for which C_{thresh}-$\bullet C \bullet C_0 < C_{thresh}$. This class of cases is represented in Figure 1 by values of C_0 that fall within region (2), and in which C_1 therefore falls within region (3) or (4). If C_0 continues to increase beyond this range to the range in which $C_0 \bullet C_{thresh}$, both the conventional speaker and the programmer will satisfy the proximate cause requirement.[175]

The "size" of this class of cases depends on the value of $\bullet C$, which represents the marginal increase in causal chain strength attributable to the fact that the speech in question is embodied in software. Given the assumptions indicated above, $\bullet C$ may be interpreted more broadly as a quantitative indicator of the marginal causal power of software. In short, the greater the causal power of software, the larger the class of cases in which software speech may satisfy the proximate cause requirement, even though conventional speech having the same expressive content would not satisfy the proximate cause requirement.

This class of cases should be of concern to civil libertarians who ascribe to the tenet that the liability outcome in a case involving highly-protected speech should not vary based merely on the form the speech takes.[176] One solution to this problem would be to increase the proximate cause threshold C_{min} (and corresponding intent threshold I_{min}) applied in software speech cases so as to equalize the outcomes in corresponding cases of conventional speech and software speech. This would eliminate any disparity in liability outcomes between cases involving software speech and cases involving conventional speech.[177] In particular, one could employ one set of thresholds in cases involving conventional speech and another, higher, set of thresholds in cases involving software speech. By making the thresholds in software speech cases sufficiently high, one could ensure that in any corresponding pair of conventional speech and software speech cases, the outcomes with respect to both the intent and proximate cause determinations would always be the same.

[175] All of the analysis just presented for proximate cause applies equally to intent, given the same assumptions. Any of the discussion herein that refers solely to proximate cause should also be interpreted as applying equally to intent.

[176] *See supra* note 169.

[177] For purposes of simplicity, the present discussion envisions the use of one set of heightened intent and causation requirements for software speech cases and another set of intent and causation thresholds for all other cases, including both those involving speech and those involving conduct. There could potentially be three sets of intent and causation requirements: (1) a first set imposed by the underlying tort or criminal claim, which would apply in cases not involving protected speech; (2) a second set of intent and causation requirements, higher than the first set, which would apply to cases involving protected *conventional* (i.e., non-software) speech; and (3) a third set of intent and causation requirements, higher than the second set, which would apply to cases involving protected software speech. Use of three sets of intent and causation requirements, however, is not necessary to demonstrate the conclusions drawn herein.

More specifically, the minimum intent and proximate cause thresholds in conventional speech cases could continue to be I_{thresh} and C_{thresh}, respectively. In software speech cases, however, the minimum intent threshold could be heightened to $I_{thresh}+\bullet I$ and the minimum proximate causation threshold could be heightened to $C_{thresh}+\bullet C$. As a result, the increased causal power of software would be precisely offset (in theory) by concomitantly increased thresholds for intent and causation, thereby ensuring[178] the same outcome in corresponding cases of conventional speech and software speech. To understand why this is true, refer again to Figure 1 and the class of cases in which C_0 is less than the proximate cause threshold C_{thresh} (i.e., values of C_0 that are in regions (1) or (2)). Since $C_1 = C_0+\bullet C$, the programmer-speaker will also fail to satisfy the heightened proximate cause threshold (i.e., $C_1 < C_{thresh}+\bullet \mathrm{C}$) in this class of cases. In other words, if the heightened proximate cause threshold $C_{thresh}+\bullet \mathrm{C}$ is applied in software speech cases, no programmer will satisfy the heightened proximate cause threshold if a corresponding conventional speaker would satisfy the normal proximate cause threshold. Similar logic justifies the more general conclusion that programmers will satisfy the heightened proximate cause threshold in only those cases where a corresponding conventional speaker satisfies the normal proximate cause threshold.

One could attempt to achieve this equalization of outcomes in cases of conventional speech and software speech by applying, perhaps, the *Sullivan* standard of intent and/or the *Brandenburg* causation requirement in cases of highly-protected conventional speech (e.g., political speech) and by adopting even higher standards in cases involving highly-protected software speech. For example, in cases involving software speech, one could require specific intent (rather than the *Sullivan* recklessness standard) and a substantial certainty of imminent harm (rather than the *Brandenburg* requirement of likelihood of harm) as the intent and proximate cause thresholds, respectively. Requiring such super-heightened standards of intent and causation would be an attempt to counterbalance the tendency for software speech to be afforded less protection than conventional speech and thereby equalize the liability outcomes.

Although some critiques of the encryption export regulations and the DMCA seem to imply that such equalization of outcomes is desirable,[179] that may not be true. The interests in freedom of expression and harm regulation may be better balanced by adopting the heightened, but not super-heightened, intent and causation requirements of *Sullivan* and *Brandenburg*, respectively, in all cases involving highly-protected speech, whether such speech is conventional speech or software speech.

[178] This statement assumes that intent and causation are the only factors that would differ in cases of conventional speech and software speech.

[179] *See supra* note 169 and sources cited therein.

The use of these standards would provide significant protection for software speech without guaranteeing the same outcome in all corresponding cases of conventional and software speech. For example, it is conceivable that the virus author in at least one of the hypotheticals set forth above could satisfy the intent standard of *Sullivan* and the proximate cause standard of *Brandenburg* even though the mechanical engineer in the corresponding magnet hypothetical would not satisfy both requirements.

Let the marginal increase in causal chain strength required by *Brandenburg* be represented by C_H and the marginal increase in intent required by *Sullivan* be represented by I_H (where "H" stands for "heightened"). Let C'_{thresh} be the heightened causation requirement proposed herein for both conventional speech and software speech cases, where $C'_{thresh}= C_{thresh}+C_H$. Similarly, let I'_{thresh} be the heightened intent requirement proposed herein for both conventional speech and software speech cases, where $I'_{thresh}= I_{thresh}+I_H$.

Even if such heightened standards were adopted, software speech would receive less protection than conventional speech if $C_H < \bullet C$ and/or if $I_H < \bullet I$. In such circumstances, the heightened intent requirement I'_{thresh} is greater than I_{thresh} but less than $I_{thresh}+\bullet I$, and (as shown in Figure 1) the heightened proximate cause requirement C'_{thresh} is greater than C_{thresh} but less than $C_{thresh}+\bullet C$. As a result, there would remain a class of cases in which programmers would satisfy the heightened intent and causation requirements I'_{thresh} and C'_{thresh} even though corresponding conventional speakers would not satisfy such requirements.

This class of cases would be characterized by values of C_1 in which $C'_{thresh} \bullet C_1 < C_{thresh}+\bullet C$, represented by region (4) in Figure 1. The size of region (4) is determined by the difference between the causal power of software ($\bullet C$) and the marginal increase in causal chain strength (C_H) required by *Brandenburg*. Although it is not possible to determine *a priori* whether the marginal increase in causal chain strength required by *Brandenburg* (C_H) is greater or less than the marginal causal power of software speech ($\bullet C$), the most interesting situation is that where $C_H < \bullet C$. In such a situation, at least some cases will fall into region (4). Similarly, the most interesting situation in the context of intent is one in which $I_H < \bullet I$, i.e., in which the heightened intent requirement required by *Sullivan* is less than the increased intent ($\bullet I$) attributable to programmers' knowledge of the causal power of software. It should not be surprising that at least some cases may fall into region (4) and that, in at least some cases, programmers may be held liable for harm caused by highly-protected software speech, even if a corresponding conventional speaker would not be held liable for conventional speech having the same expressive content. The combination of software and the Internet make software speech more powerful than conventional speech and, under conventional legal principles, with this increased great power may come great responsibility and increased liability.

VI. CONCLUSIONS AND RECOMMENDATIONS

The analysis above indicates that many challenges lie ahead for those who seek to protect freedom of expression in software. The following sections further detail some of these challenges and my recommendations for fixing them.

A. The Challenges Ahead

1. Regulation of Uncontroversial Harms

It has been easy for civil libertarians to take the moral high ground in the cases which have drawn the most attention so far, such as the cryptography and DMCA cases, because the regulations at issue have been based on policy objectives that are objectionable on grounds unrelated to the speech of programmers. Future cases may be more morally ambiguous, at least in part because they will lack a clearly identifiable enemy. Cases such as those involving the benevolent virus author are not particularly far-fetched, especially when it is recognized more generally that various kinds of software distributed by computer security professionals for educational or other legitimate purposes may also be used to cause damage to computers or perform other mischief. It will be more difficult to challenge the application of laws that prohibit causing damage to computers or distributing software for the purpose of causing damage to computers because the government interest in prohibiting these harms is much stronger and clearer than in the cases involving the encryption export regulations and the DMCA.

2. Software Speech and Content-Neutral Laws

Overall, generally-applicable content-neutral laws will likely continue to encroach more closely upon protected software speech as software becomes more powerful and more widely used to achieve harmful as well as beneficial goals, thereby making acts performed using software increasingly susceptible to violating the law. It will be increasingly difficult to bring legal challenges as well as rally political opposition to generally-applicable laws that target uncontroversial harms and that have only an incidental impact on the speech of programmers. A federal statute prohibiting the distribution or use of computer viruses simply does not have the taint of censorship associated with seditious libel statutes[180] or statutes regulating indecent speech,[181] even when such a statute is applied to a programmer who sincerely intends to express ideas

[180] *See generally* Abrams v. United States, 250 U.S. 616 (1919); Debs v. United States, 249 U.S. 211 (1919); Schenk v. United States, 249 U.S. 47 (1919).

[181] *See* Reno v. ACLU, 521 U.S. 844 (1997).

through source code. Because such statutes are content-neutral, they will continue to receive, at most, intermediate scrutiny under the First Amendment.[182] It is therefore incumbent on those concerned with freedom of expression to develop arguments that apply to software speech within the rubric of intermediate scrutiny. Such arguments are most likely to be successful in cases involving statutes that are written more broadly than necessary. However, in the case of statutes that are well drafted and narrowly tailored to focus on harms which the government has a legitimate interest in prohibiting, the best arguments will focus on the manner in which the applicable statute is applied to the particular defendant in the specific case, as described in more detail below.

This raises the more general question–the answer to which is beyond the scope of this paper–of whether the doctrine of content-neutrality requires reconsideration.[183] Software speech that expresses scientific theories or educational information represents perhaps the first entire category of core protected speech of which regulations are destined to be judged according to intermediate scrutiny, rather than strict scrutiny. In the past, the only statutes having terms encompassing the distribution of educational materials by academics or professionals were those that targeted speech based on its content or viewpoint and which would therefore be subject to strict scrutiny. Distribution of educational information was extremely unlikely to fall within the purview of any content-neutral laws. The particular characteristics of software speech, however, make it especially susceptible to indirect regulation by statutes that do not target speech on their face.

One possible solution to this problem is adoption of strict scrutiny review in cases involving software speech, even when the applicable statute does not facially target speech.[184] In the context of the DMCA,

[182] *See supra* note 82.

> The principal inquiry in determining whether a statute is content-neutral is whether the government has adopted a regulation of speech because of agreement or disagreement with the message it conveys. The government's purpose is the controlling consideration. Here, the parties have pointed to no portion of the legislative history that demonstrates a congressional intent to target speech because of its expressive content. Rather, Congress sought ways to further electronic commerce and protect property rights, while at the same time protecting fair use. In order to balance these priorities, Congress sought to ban trafficking in any technology or device that could be used to circumvent technological restrictions that served to protect the rights of copyright owners.

United States v. Elcom Ltd., 203 F. Supp. 2d 1111, 1128 (N.D. Cal. 2002).

[183] For a discussion of the application of the First Amendment in the context of content-neutral regulations, see generally David S. Day, *The Incidental Regulation of Free Speech*, 42 U. MIAMI L. REV. 491 (1988); Kagan, *supra* note 82; Geoffrey R. Stone, *Content-Neutral Restrictions*, 54 U. CHI. L. REV. 46 (1987); Stone, *supra* note 82.

[184] Such an approach is not without precedent. Professor Elena Kagan, for example, argues that "First Amendment law, as developed by the Supreme Court over the past several decades, has as its primary, though unstated, object the discovery of improper governmental motives. The doctrine comprises a series of tools to flush out illicit motives and to invalidate actions infected with them." Kagan, *supra* note 82, at 414. Professor Kagan further observes that:

for example, the argument for applying strict scrutiny would be as follows: Congress, in regulating the distribution of "devices" that circumvent access- or copy-control technologies must have known that (a) such devices would most frequently be implemented in software; and (b) software is speech. Therefore, Congress intended the DMCA to target software speech, even though the text of the statute refers only to "devices" and not to speech.[185]

3. Increasing Technological Sophistication of Legal Actors

Many arguments against particular laws regulating software speech either imply or expressly assert that such laws are motivated by stupidity, technological illiteracy, or an affirmative intent to censor on behalf of the legislature, the judiciary, law enforcement officers, or private litigants.[186]

> [A] generally applicable law by definition targets not a particular idea, nor even ideas broadly speaking, but an object that need not, and usually does not, have any association with ideas whatsoever . . . . The breadth of [most content-neutral] laws makes them poor vehicles for censorial designs; they are instruments too blunt for effecting, or even reflecting, ideological disapproval. Thus, incidental restrictions receive minimal scrutiny because of the likelihood that they also will be accidental restrictions in the relevant sense—that they will result from a process in which officials' hostility (or partiality) toward ideas played no role.
>
> . . . [W]hen a proposed law, by its terms, focuses on nonexpressive conduct, restricting speech only as an incidental and thus a covert matter, the probability increases that a legislator will consider the regulation divorced from hostility or sympathy toward particular messages.

Id. at 495–96. Professor Kagan notes, however, that "[a] law of generally application . . . may have such dramatic—and apparent—effects on expressive activities that it might as well target those activities in express terms." *Id.* at 496. At least some content-neutral regulations may fall into this category with respect to their application to software speech. For example, in *Arcara v. Cloud Books, Inc.*, the U.S. Supreme Court explained that generally applicable laws warrant First Amendment scrutiny "only where it was conduct with a significant expressive element that drew the legal remedy in the first place, . . . or where a statute based on a nonexpressive activity has the inevitable effect of singling out those engaged in expressive activity." 478 U.S. 697, 706–07 (1986). Generally applicable statutes may satisfy one or both of these criteria when applied to protected software speech, thereby providing a basis for First Amendment scrutiny of such statutes where otherwise none would be given.

[185] *See, e.g.*, Tien, *supra* note 6, at 695–96 ("[T]o the extent that functionality must fit into the rubric of 'content-based' vs. 'content-neutral' regulation, functionality-based regulation is much closer to content-based than content-neutral regulation for the simple reason that the content of software is inextricable from its functionality"). Fox argues:

> One problem with the functionality argument is that it tries to split an undifferentiated whole, just as Professor Ely suggests the *O'Brien* court did. There is no possible way for the function of computer code to be removed from its expressive nature: It is meant to do a purpose. Because of this, it seems quite hard for the government to claim that a regulation that involves the function of a piece of code is unrelated to expression. This is largely ignored by those that would still apply *O'Brien*, holding the word "functionality" as a talisman against claims that the regulations are content-based and ignoring its meaning in this context.

Fox, *supra* note 102, at 908.

[186] For example, Professor David Touretzky argues that:

> [T]he most recent attempt to limit free speech comes from Hollywood [in the form of the DMCA] . . . [a]nd it's directed at computer programmers . . . . [S]ince 1909 Congress has been enacting whatever copyright provisions the lawyers for the major content-producing and distributing industries negotiate among themselves.

Touretzky, *supra* note 25, at 23. In these few sentences, Professor Touretzky accuses Congress, the content industry, and the content industry's lawyers of intentionally attempting to stifle the speech of computer scientists. The Appellee's Brief in *Bernstein IV* argues:

> [T]he regulations have more than a close enough nexus to speech to pose risk of censorship, they directly restrict scientific speech in a particular subject area of applied mathematics, specifically

In particular, the arguments made by the "software-as-speech" camp[187] imply that the laws being challenged would surely be held unconstitutional if only the courts would come to understand that software, particularly in source code form, is used by human programmers to express ideas to each other.

Although these criticisms may be valid to a certain extent, they are of limited significance for at least two reasons. First, even if such ignorance or malice is an actual cause of regulation, it is not the fundamental cause. Rather, as illustrated above, thorny problems involving the regulation of software speech would still exist even if all actors in the legal system were immediately transformed into civil libertarian computer science professors. Second, technological literacy within the legal system is likely to increase over time. Civil libertarians are increasingly likely to confront legislators, prosecutors, judges, and private litigants who understand how computer technology works and who, therefore, are able to tailor their actions and arguments to be immune from this particular criticism.

4. Software and Actual Harm

The virus hypotheticals and other examples described above are cases in which the distribution of software has caused actual harm, such as the destruction of electronic data. Both the encryption export regulations and the DMCA, however, prohibit the distribution of software without requiring the distribution to have caused any actual harm. The encryption export regulations, for example, prohibit the distribution of encryption software without obtaining prior permission from the government.[188] These regulations prohibit the distribution of software even before such software has caused any actual harm. Similarly, the DMCA does not require that the plaintiff prove actual damages resulting from the distribution of a circumvention device; the mere act of distributing, or "trafficking in," circumvention devices is sufficient to violate the terms of the statute.[189] This is one of the primary grounds on which the encryption export regulations and the DMCA have been criticized.[190]

> prevent such speech on the Internet, and restrict private communication. Further, the regulations grant unfettered discretion to the bureaucrats who implement it, and lack the procedural safeguards required by Freedman v. Maryland, 380 U.S. 51 (1965). The lack of narrow, definite and objective standards causes self-censorship and permits unreviewable content-based discrimination, both of which are demonstrated in the record.

Appellee's Brief, part IV, *Bernstein IV* (No. 97-16686), *supra* note 13.

[187] *See infra* Part I.B.

[188] *See generally Bernstein I*, 922 F. Supp. 1426 (N.D. Cal. 1996).

[189] *See supra* note 11 and accompanying text.

[190] *See, e.g.*, Brief of Amicus Curiae Garrett Epps, *Bernstein IV* (No. 97-16686) ("The EAR amendments . . . do not attempt to control speech that actually incites imminent lawless action. Instead, they endeavor to contain academic dialogue because it supposedly enables actions that could

In First Amendment terms, the prohibition of speech before it has occurred is referred to as a "prior restraint."[191] Along the continuum of First Amendment protection, speech receives the highest degree of protection against prior restraints,[192] significant but lesser protection when the speech itself is targeted absent any actual harm,[193] and yet less protection when the speech has actually caused harm.[194] This continuum of protection mirrors the extent of the burden that each kind of regulation places on speech.

Prior restraints, whether in the form of statutes or executive actions, have been relatively rare in the last century.[195] Moving down the

assist illegal activity. This connection is far too tenuous to pass the Brandenburg threshold."). The Defendant-Appellant brief in *Corley* argues:

> According to the District Court, because the speech at issue here can be moved easily throughout the Internet and might harm the Studios' commercial interests, traditional rules of causation should be abandoned in order to enjoin the press for what its readers might do. This analysis contradicts First Amendment law and tradition.
>
> . . . .
>
> . . . [T]his case does not involve any copying of copyrighted expression. In the absence of any copying, the District Court was forced to hold that the 'injury' element required by § 1201 was satisfied by the mere publication of "the readily available means of circumventing the CSS access control system on their DVDs." Thus, because the Studios could not show either direct or indirect harm from publication of DeCSS, the District Court effectively held that because DeCSS was publicly disseminated on the Internet, which reaches an enormous audience, the feared harm was *likely* to result.

Brief for Defendant-Appellant, Part III.A, Universal City Studios, Inc. v. Corley, 273 F.3d 429 (2d Cir. 2001) (No. 00-9185), *supra* note 116 (citation omitted).

[191] One court has, in fact, held the encryption export regulations to constitute a prior restraint in violation of the First Amendment. *Bernstein III*, 974 F. Supp. 1288, 1310 (N.D. Cal. 1997). In Plaintiff's Motion for Preliminary Injunction in *Junger*, the attorneys argued that because of insufficient First Amendment protection, proper speech was chilled:

> The administrative licensing scheme in International Traffic in Arms Regulations (ITAR), which requires a license before exporting any type of cryptographic software no matter how trivial, constitutes the clearest and most blatant form of prior restraint . . . . Prof. Junger's speech has been, and continues to be, chilled to the point of freezing. He has refrained from teaching his class to foreign students, discussing his work with foreign colleagues and publishing material that contains cryptographic information. . . . The present version of ITAR continues to reach communication that is protected by the First Amendment and that does not pose a threat to national security.

Brief In Support of Plaintiff's Motion for Preliminary Injunction, Part II.B, Junger v. Daley, 8 F. Supp. 2d 708 (N.D. Ohio 1998) (No. 96-CV-1723).

[192] The Supreme Court has held that "[a]ny system of prior restraint comes to this Court bearing a heavy presumption against its constitutional validity." Bantam Books, Inc. v. Sullivan, 372 U.S. 58, 70 (1963). The Court has recognized only a narrow number of situations in which prior restraints might be permissible, such as restraints necessary to protect against imminent threats to national security. *See* United States v. Progressive, Inc., 467 F. Supp. 990, 992 (W.D. Wis. 1979); New York Times Co. v. United States, 403 U.S. 713, 726 (1971). For an overview of the First Amendment as it applies to prior restraints, see SMOLLA & NIMMER, *supra* note 37, § 15.

[193] The *Brandenburg* standard, for example, applies in circumstances where a speaker has spoken, but in which no harm has yet been caused.

[194] *See, e.g.*, *Bernstein II,* 945 F. Supp. 1279, 1286 (N.D. Cal. 1996) ("It is axiomatic that the First Amendment is more tolerant of subsequent criminal punishment of speech than it is of prior restraints on the same speech."). This paper focuses on cases involving subsequent punishment, exemplified by the virus hypotheticals.

[195] For twentieth-century cases addressing prior restraints, see generally Nebraska Press Ass'n v. Stuart, 427 U.S. 539 (1976); *New York Times.,* 403 U.S. at 713; Near v. Minnesota, 283 U.S. 697 (1931); *Progressive, Inc.*, 467 F. Supp. at 996 (noting that "[i]f a preliminary injunction is issued, it will

continuum, laws regulating speech or the distribution of devices once such speech or distribution has occurred, but in the absence of actual harm, have been much more common. These regulations typically have been limited to speech or devices that are viewed as being particularly susceptible to causing harm.[196] Moving to the end of the continuum, laws regulating speech and devices that have actually caused harm are commonplace and are the subject of most tort and criminal law causes of action that are applicable to speech.[197]

The prospective aspect of the DMCA, however, may be a harbinger more generally of content-neutral laws that prohibit the mere distribution of devices for performing particular functions, particularly when such devices are typically embodied in software and/or perform functions over the Internet. The logic behind this trend upward in regulation is that it is but a short step from the distribution of such items to their use and redistribution. At least one court has adopted this reasoning in support of the DMCA:

> There was a time when copyright infringement could be dealt with quite adequately by focusing on the infringing act. If someone wished to make and sell high quality but unauthorized copies of a copyrighted book, for example, the infringer needed a printing press. The copyright holder, once aware of the appearance of infringing copies, usually was able to trace the copies up the chain of distribution, find and prosecute the infringer, and shut off the infringement at the source.
>
> In principle, the digital world is very different. Once a decryption program like DeCSS is written, it quickly can be sent all over the world. Every recipient is capable not only of decrypting and perfectly copying plaintiffs' copyrighted DVDs, but also of retransmitting perfect copies of DeCSS and thus enabling every recipient to do the same. They likewise are capable of transmitting perfect copies of the decrypted DVD. The process potentially is exponential rather than linear. . . .
>
> . . . .
>
> These considerations drastically alter consideration of the causal link between dissemination of computer programs such as this and their illicit use. Causation in the law ultimately involves practical policy judgments. Here, dissemination itself carries very substantial risk of imminent harm because the mechanism is so unusual by which dissemination of means of circumventing access controls to copyrighted works threatens to produce virtually unstoppable

constitute the first instance of prior restraint against a publication in this fashion in the history of this country, to this Court's knowledge.").

[196] The "clear and present danger" doctrine (first announced in *Schenk v. United States*, 249 U.S. 47 (1919)) and the "fighting words" doctrine (first announced in *Chaplinsky v. New Hampshire*, 315 U.S. 568 (1942)) may be seen as attempts to define the kinds of speech that are particularly susceptible to causing harm.

[197] *See, e.g.*, *supra* note 89 and accompanying text.

> infringement of copyright. In consequence, the causal link between the dissemination of circumvention computer programs and their improper use is more than sufficiently close to warrant selection of a level of constitutional scrutiny based on the programs' functionality.[198]

One may disagree with the application of this logic to the copyright context but still appreciate the validity of the factual observations it contains with respect to the nature of software and the Internet. The fact that software may be copied, distributed, and executed much more quickly and easily than previous forms of technology may lead lawmakers to continue to enact legislation that targets the mere distribution of software and other technologies rather than their use. These same features of software and the Internet may lend support to the legal conclusion that such measures are necessary to effectively regulate the targeted harms. The challenge for civil libertarians is to craft arguments that justify narrowing such regulations to minimize their impact on freedom of expression. They must also recognize the validity of the factual finding implicit in such regulations that technology may be distributed and used at lower cost in the digital world than was previously possible in the non-digital world.

B. First Amendment Defenses Must Rise to These Challenges

Those concerned with civil liberties have their work cut out for them. In this section, I recommend strategies for civil libertarians to adopt if software speech is to obtain strong First Amendment protection.

1. Accept the Functionality of Software

Those in the "software-as-speech" camp often deny that software, particularly in source code form, is functional. Civil libertarians put forth arguments that usually refuse to admit or are unable to recognize that source code can be harmful. Such arguments assume that to recognize the ability of source code to cause harm would be to forfeit the ability to argue for First Amendment protection and to argue against the suppression of source code. In other words, the *factual* conclusion that software is powerful is rejected because the predicted, and feared, *legal* outcome—regulation of software speech—is viewed as unacceptable.[199]

[198] Universal City Studios, Inc. v. Reimerdes, 111 F. Supp. 2d 294, 331–32 (S.D.N.Y. 2000).

[199] For example, the argument is frequently made that source code cannot be a "device" because the expressive content of source code may alternatively be expressed in other forms, such as in English prose and poetic verse. According to this argument, such forms of expression are clearly not "devices," and, therefore, source code is not a device. For example, Professor David Touretzky maintains a "Gallery of DeCSS Descramblers" on a Web site. *Gallery of DeCSS Descramblers*, *at* http://www-2.cs.cmu.edu/~dst/DeCSS/Gallery/index.html (last visited Apr. 15, 2004). The Web site includes a collection of instructions for breaking the Content Scramble System ("CSS") encryption scheme that is used to encrypt digital versatile discs ("DVDs"). *Id.* The instructions are embodied in

Such a results-oriented approach is both unnecessary and inconsistent with the undisputable fact that software can cause harm.[200] The mere fact that a particular kind of speech is powerful and can cause harm–even significant harm that is appropriately subject to regulation by law–is not a sufficient reason for making First Amendment protection unavailable for such speech. Both political speech and religious speech can be extremely powerful and can be used to accomplish either benevolent or malevolent ends, yet both enjoy strong First Amendment protection. Protection for such speech need not be premised on its lack of power; to the contrary, the fact that political speech can be powerful is one reason to protect it vigorously.[201] Protection may be premised on the recognition that humans, particularly human governments, are neither infallible nor omniscient, and that the best protection against oppressive government is a marketplace of ideas in which political ideas—even powerful and harmful political ideas—can compete against each other.[202]

Arguing for strong First Amendment protection of software speech does not require denying that software is functional or that software can cause harm. Indeed, if software speech is not powerful, why do we care about it so much? Rather, the strongest and most lasting arguments for First Amendment protection of software speech will wholeheartedly

various forms, including source code, songs, poetry, and English-language descriptions. *Id.* The Web site was created, in part, "to point out the absurdity of Judge Kaplan's position [in Universal City Studios v. Reimerdes, 82 F. Supp. 2d 211(S.D.N.Y. 2000)] that source code can be legally differentiated from other forms of written expression." *Id.* Professor Touretzky provides a more detailed argument for his position in *Free Speech Rights for Programmers, supra* note 25, at 24.

Professor Touretzky's argues that the examples on his Web site demonstrate the absurdity of Judge Kaplan's position. One way to interpret this argument is that Professor Touretzky is drawing on an intuition that instructions embodied in forms such as songs and English-language descriptions cannot be devices. One problem with this argument is that the term "device" carries so much connotative baggage (it connotes, for example, moving electromechanical parts) that it may not be possible to have a meaningful discussion about whether new kinds of entities such as source code embodied in paper or electrical signals fall within the definition of "device." If one asks, instead, what features of a device or substance, such as firearms, make its distribution properly subject to regulation by law, it is not unreasonable to answer: the ability to cause significant harm relatively directly, easily, and automatically. The nature of modern digital computers and their widespread availability would make a T-shirt on which novel virus source code is printed satisfy this definition without having to answer the question of whether such a T-shirt is a "device." Certainly the examples in the "Gallery of DeCSS Descramblers" cannot be intended to support the general proposition that speech cannot be regulated merely because it is expressible in a small number of words, in code, in song, or on a T-shirt. After all, the current coordinates of military troops may be expressed in all of these forms, yet posting such numbers in wartime would not be protected by the First Amendment. Professor Touretzky's examples, therefore, while thought-provoking, do not by themselves explicate any particular rule for distinguishing protected speech from unprotected speech.

[200] Computer viruses are perhaps the best-known example of computer programs that can cause harm, but they are certainly not the only type of malicious software that has been created.

[201] Justice Brandeis' famous statement in *Whitney* that, in general, "the remedy to be applied [to false or otherwise harmful speech] is more speech," provides one justification for granting strong protection to speech precisely because of its power to counterbalance falsehoods and other harms caused by speech. Whitney v. California, 274 U.S. 357, 377 (1927).

[202] *See* Abrams v. United States, 250 U.S. 616, 630 (1919) ("[T]he best test of truth is the power of the thought to get itself accepted in the competition of the market."); *Whitney*, 274 U.S. at 376 ("If there be time to expose through discussion the falsehood and fallacies, to avert the evil by the processes of education, the remedy to be applied is more speech, not enforced silence.").

accept the functionality of software. It is precisely because source code has the power to control computers that computer programmers use source code to design and implement software. Government and private industry would be unlikely to invest heavily in software development if the written output of programmers had no power to control machines, but rather merely represented idle expressions of the programmers' ideas.[203] All of the benefits claimed for the software development process in comparison to the hardware development process, such as decreased cost and shorter development times, are predicated on the premise that writing source code is not an idle enterprise intended merely to express ideas. Rather it is understood as an engineering design process intended to produce source code having causal powers equivalent to the hardware for which it serves as an alternative.[204]

Rather than argue that source code is powerless, civil libertarians should explicitly recognize and even embrace the power of source code just as we embrace the power of political and religious speech. Failing to do so is not to forfeit the battle for First Amendment protection or to renounce one's civil libertarian credentials; rather, such a recognition merely serves to move the debate onto the same plane as the traditional debate over First Amendment protection for political speech. Placing it in this arena allows civil libertarians to hold the moral, political, and legal high ground. Arguments in favor of First Amendment protection for source code should mirror the classic arguments for protection of political speech and thereby strive to demonstrate that source code should be protected in spite of the possibility that it may be harmful.[205]

[203] For example, Hamming argues:

> At the heart of computer science lies a technological device, the computing machine. Without the machine almost all of what we do would become idle speculation, hardly different from that of the notorious Scholastics of the Middle Ages. The founders of the Association of Computing Machinery ("ACM") clearly recognized that most of what we did, or were going to do, rested on this technological device, and they deliberately included the word "machinery" in the title [of the ACM]. There are those who would like to eliminate the word, in a sense to symbolically free the field from reality, but so far these efforts have failed. I do not regret the initial choice. I still believe that it is important for us to recognize that the computer, the information processing machine, is the foundation of our field.

R.W. Hamming, *One Man's View of Computer Science,* 16 J. ASS'N COMPUTING MACHINERY 3, 5 (1969).

[204] Samuelson, *supra* note 136, at 2316 ("The engineering designs embodied in programs could as easily be implemented in hardware as in software**,** and the user would be unable to distinguish between the two."); Samuelson, *supra* note 25, at 1130–33 ("There is no fixed dividing line between computer programs and computers because anything that can be implemented in software can also be implemented in hardware.").

[205] Of course, source code should also be protected in particular cases in which the source code at issue has caused no actual harm, or poses less than a certain threat to cause harm in the future, regardless of the potential of source code in general to cause harm.

2. Focus on Intent, Causation, and Other Claim Elements in Each Case

a. Argue for the Applicability of Heightened Requirements

As described above, facial challenges to content-neutral statutes that apply to software speech are likely to fail unless Congress drafts such statutes so broadly that they prohibit speech even when doing so is not necessary in regulating the harms to which the statutes are targeted.[206] Facial challenges to the DMCA,[207] which targets a harm that is very controversial, have yet to succeed, and only one court has sustained a facial challenge to the encryption export regulations.[208] Similar challenges are likely to be even less successful when applied to statutes that prohibit uncontroversial harms, particularly if such harms are *malum in se*.[209] Civil libertarians should abandon the sweeping

[206] The encryption export regulations, for example, were worded particularly broadly, which may explain why they fared worse than the DMCA in the courts.

[207] *See, e.g.,* United States v. Elcom, 203 F. Supp. 2d 1111, 1133 (N.D. Cal. 2002) (rejecting defendants' overbreadth challenge to the DMCA on the ground that "facial attacks on overbreadth grounds are limited to situations in which the statute or regulation by its terms regulates spoken words or expressive conduct").

[208] *Bernstein I,* 922 F. Supp. 1426, 1439 (N.D. Cal. 1996) (denying government's motion to dismiss); *Bernstein II,* 945 F. Supp. 1279, 1296 (N.D. Cal. 1996) (granting plaintiff's motion for summary judgment in part). *But see* Karn v. U.S. Dep't of State, 925 F. Supp. 1, 14 (D.D.C. 1996) (rejecting facial challenge to encryption export regulations); Junger v. Daley, 8 F. Supp. 2d 709, 724 (N.D. Ohio 1998) (rejecting facial challenge to encryption export regulations).

[209] One possibility, not pursued in-depth here, would be to argue for different standards to apply in cases involving actions that are *malum in se* than in cases involving actions that are merely *malum prohibitum*. One might, for example, argue for strict scrutiny to apply to statutes that prohibit actions that are *malum prohibitum* but accept intermediate scrutiny in cases involving statutes that prohibit actions that are *malum in se*. One could further subdivide the standards to be applied based on the kind and degree of harm sought to be protected by a particular cause of action, or the kind and degree of harm actually caused in a particular case. This would comport with the tort law principle that the degree of care required is graduated according to the danger attendant on the activity in which the defendant engages; the greater the danger, the greater the degree of care required. 57 AM. JUR. 2d *Negligence* § 154 (2000).

More generally, one might apply a duty calculus which balances foreseeability against factors including the extent of the burden to be imposed on the defendant so that, in cases where the burden imposed on speech would be great, a high degree of foreseeability would be required. Conversely, in cases in which the interest in preventing the harm at issue is particularly strong, or there are readily available and non-burdensome means for preventing the harm, a lesser degree of foreseeability would be required. *See* Nicole M. v. Sears, Roebuck & Co., 90 Cal. Rptr. 2d 922 (Cal. Ct. App. 1999); Sims, *supra* note 93, at 276 (discussing the court's application of a balancing test which incorporates First Amendment interests into the balancing test announced by Judge Learned Hand in *United States v. Carroll Towing Co.*, 159 F.2d 169 (2d Cir. 1947)).

In *Rice*, another noteworthy approach that has been proposed is to adopt the "detail for imminence" trade. Rice v. Paladin Enter., Inc., 128 F.3d 233, 246 (4th Cir. 1997). Smolla argues that according to this decision:

> [m]aterials intended to provide detailed training instruction for planning and executing crimes deserve no First Amendment protection even though there may be a gap in time between the provision of the assistance and the perpetration of the crime, because the detailed training provides a causal nexus to lawless action that serves as a substitute for the nexus normally required in time.

Rodney A. Smolla, *Should the* Brandenburg v. Ohio *Incitement Test Apply in Media Violence Tort Cases?*, 27 N. KY. L. REV. 1, 43 (2000).

argument that "software = speech" and should therefore receive the highest degree of First Amendment protection, because such an argument has no basis in law. After all, the fact that "talking = speech" does not imply that all talking should receive the highest degree of First Amendment protection. It would be inconsistent to apply a variety of First Amendment standards to conventional speech based on factors such as the speaker's intent and the harm to be prevented, and yet to apply a single overarching standard of extremely strong protection for all software speech, regardless of the surrounding circumstances. If software is a kind of speech and the applicability of the First Amendment to speech in general varies depending on the circumstances, then *a fortiori* the applicability of the First Amendment to software speech should vary according to the same rules that apply to conventional speech. Factors that should be taken into account to determine the degree of protection to be afforded to software speech in a particular case include the intent of the speaker, the likely or actual (beneficial and harmful) results of the speech, the foreseeability of such results, and the extent to which alternative avenues of expression are available to the speaker.[210] The need to tailor the kind and degree of protection to the circumstances of a particular case will increase as software becomes capable of performing a wider variety of functions, becomes accessible to a wider variety of people, and becomes capable of causing a wider variety of beneficial and harmful results.[211]

Instead of arguing for a single overarching standard of First Amendment protection to apply to software speech, civil libertarians should argue for the application of heightened intent and proximate cause requirements in cases involving protected software speech,[212] such

[210] *See* Sims, *supra* note 93.

[211] As computer viruses and computer security flaws cause increasing damage, and as consumers, government, and the private sector grow increasingly unwilling to accept the damage that actually results from faulty or malicious software as an ordinary cost of doing business, increasing numbers of lawsuits resulting from such harm are likely to arise. There are strong signs that software's liability honeymoon is drawing to a close as consumers and legislators lose their patience with bug-ridden and insecure software. *See, e.g.,* Charles C. Mann, *Why Software is So Bad*, TECH. REV., July - Aug. 2002, at 33 (describing increasing public and private frustration with faulty and insecure software and predicting that product liability lawsuits based on faulty lawsuits are not far away); Reuters, *Software Errors Cost Billions* (June 28, 2002) *available at* http:// www.accordsqa.com/news/ 020628_costbillions.html (summarizing a U.S. Department of Commerce study finding that software bugs cost the U.S. economy about $59.5 billion annually); Evan Hansen, *Your PC's Enemy Within*, CNET NEWS.COM (June 24, 2002), *at* http://news.com.com/2009-1023-937457.html ("The Wild West days of cyberspace are over—and, like it or not, it's time for government to change its laissez-faire attitude toward the Internet and create laws that clearly prevent unscrupulous businesses from preying on unsuspecting consumers and seizing control of computers."); David Coursey, *Gates's E-mail Edict: How it Marks the End of Innocence*, ANCHORDESK (January 17, 2002), *at* http://reviews-zdnet.com.com/4520-6033_16-4206850.html (describing Bill Gates' memo declaring that Microsoft programmers should make software security their highest priority as a statement that "computing, which used to be a small town where people were nice to one another and you could leave your keys in your car, has become a big city that's also inhabited by people who are, well, evil").

[212] It may also be desirable to argue for heightened duty requirements, although that avenue is not pursued herein.

as the requirements embodied in *Brandenburg* and/or *Sullivan*.[213] The application of the *Brandenburg* and *Sullivan* requirements to software speech cases in which no actual harm has yet been caused would result in requirements that: (1) the programmer-speaker act recklessly with respect to the possibility that a third party would be harmed by the speech; and (2) the software speech at issue be directed to producing, and be likely to produce, imminent unlawful action capable of causing the alleged future harm. In software speech cases involving actual harm alleged to have been caused by software speech, the plaintiff would further be required to demonstrate that publication of the software speech was the proximate cause of the alleged harm. The application of even more rigorous standards than those adopted in *Brandenburg* and *Sullivan* may be appropriate in cases involving particularly valuable speech.

Although many courts have applied the *Brandenburg* standard in tort and criminal cases where the speech of the defendant forms the basis for the claim,[214] and although commentators sometimes assume or argue that *Brandenburg* is broadly applicable to any case involving speech,[215] the applicability of *Brandenburg* in all cases is far from a foregone conclusion.[216] By its very language, the *Brandenburg* standard only

[213] For a proposal that a heightened evidentiary standard be applied to the admissibility of evidence of conduct protected by the First Amendment, see generally Robert P. Faulkner, *Evidence of First Amendment Activity at Trial: The Articulation of a Higher Evidentiary Standard*, 42 UCLA L. REV. 1 (1994).

[214] *See, e.g.,* Herceg v. Hustler Magazine Inc., 814 F.2d 1017, 1023 (5th Cir. 1987) (applying the *Brandenburg* standard and finding no liability because the defendant's speech did not constitute "incitement"); Zamora v. CBS, Inc., 480 F. Supp. 199, 206 (S.D. Fla. 1979) (applying the *Brandenburg* standard and finding no liability because the defendant's speech did not constitute "incitement"); McCollum v. CBS, Inc., 249 Cal. Rptr. 187, 193 (Cal. Ct. App. 1988) (applying the *Brandenburg* standard and finding no liability because of lack of intent to cause imminent lawless action); Yakubowicz v. Paramount Pictures Corp., 536 N.E.2d 1067, 1071 (Mass. 1989) (applying the *Brandenburg* standard and finding no liability because the contents of the movie "The Warriors" did not constitute "incitement"); Davidson v. Time Warner, Inc., No. V-94-006, 1997 U.S. Dist. LEXIS 21559, at *63–71 (S.D. Tex. 1997) (recording artist held not liable under the *Brandenburg* standard for murder of police officer committed by an assailant who had allegedly been listening to a song recorded by the recording artist, which described the commission of violence against police officers); DeFilippo v. NBC, 446 A.2d 1036, 1042 (R.I. 1982) (broadcasting company held not liable under the *Brandenburg* standard for hanging death caused by imitation of a hanging stunt broadcast by the broadcasting company on a television show); Olivia v. NBC, 178 Cal. Rptr. 888 (Cal. Ct. App. 1981) (holding defendant not liable under the *Brandenburg* standard for sexual assault committed in a manner like that depicted in a film broadcast by defendant broadcasting company).

[215] *See, e.g.,* Justine Wellwood, *Tort Liability of the Media,* 15 ST. JOHN'S J. LEGAL COMMENT. 187, 217 (2000) (arguing that *Brandenburg* should be broadly applicable to "cases involving physical injuries that allegedly arise from media works" even if such speech is not political or social speech).

[216] For an excellent analysis of the applicability (or lack thereof) of *Brandenburg* in media violence tort cases, see Smolla, *supra* note 209, at 12. In general, Professor Smolla argues that *Brandenburg* was decided in the context of advocacy of violent action directed to a crowd and should be limited to cases involving such circumstances, while strongly arguing against the adoption of *Brandenburg* as an all-encompassing doctrine of First Amendment protection for speech:

> The *Brandenburg* case is an important First Amendment landmark, but it is not the only First Amendment landmark and does not restate the legal doctrine applied in all First Amendment contexts. To the contrary, modern First Amendment law is a complex maze of doctrinal formulas employing specific standards that have been tailored to particular topics of speech, modes of legal

applies to state statutes that prohibit "advocacy."[217] *Brandenburg,* therefore, may not apply to cases in which the speech at issue does not constitute "advocacy," such as the virus and magnet hypotheticals described above.[218] The reasoning in the *Rice v. Paladin* decision, for example, suggests that the *Brandenburg* standard only applies in cases involving political or social speech.[219] More generally, in cases where a negligence claim is brought against a programmer who has distributed software that allegedly has caused harm to a third party, such distribution is likely not to constitute "advocacy" and therefore would not trigger the application of the *Brandenburg* standard.[220] A number of courts have

> liability, and social contexts. There are innumerable other First Amendment contexts in which the *Brandenburg* standard just does not apply, contexts in which the Supreme Court has fashioned special standards suited for the balance of interests at hand. Because free speech issues arise in an extraordinarily wide range of circumstances and settings, the Supreme Court has not attempted to jam all free speech analysis into the "incitement" standard of *Brandenburg*, but rather has employed *Brandenburg*-style analysis only in cases dealing with *Brandenburg*-like settings.

Id.

In general, case law does not reflect any particularly coherent theory about the applicability of the First Amendment to tort or criminal law claims. Sims has noted in connection with claims for physical injuries allegedly caused unintentionally by media speech that:

> When confronted with these media physical injury cases, many courts have accepted the argument that the media defendants deserved First Amendment protection beyond the common law and statutory defenses normally afforded negligence tort defendants under state law. However, in part because the fact patterns of the cases defied easy classification, and in part because First Amendment principles themselves have often seemed murky, undeveloped or in transition, a consistent and satisfactory jurisprudence has failed to develop in this area. More often than not, the courts seem to be reaching an appropriate result, but often by following questionable constitutional logic that may only lead to greater confusion in future cases. Worse, a court will occasionally reach what seems to be an unfair result under what it believes to be, perhaps incorrectly, constitutional compulsion.

Sims, *supra* note 93, at 233–34. For additional discussion regarding why *Brandenburg* might not be applicable to cases not involving political advocacy, see *id.* at 255–62.

[217] The *Brandenburg* court stated:

> [T]he constitutional guarantees of free speech and free press do not permit a State to forbid or proscribe advocacy of the use of force or of law violation except where such advocacy is directed to inciting or producing imminent lawless action and is likely to incite or produce such action.

Brandenburg v. Ohio, 395 U.S. 444, 447 (1969).

[218] *See* Weingarten, *supra* note 93, at 747–49 (arguing that non-libelous negligent statements that result in physical harm to third parties do not "advocate" harm within the meaning of *Brandenburg* and therefore should not be analyzed according to the *Brandenburg* standard). *See also* Sims, *supra* note 93, at 260.

[219] *See* Wellwood, *supra* note 215, at 216. In addition, Smolla asserts:

> What the Court of Appeals [in the Hit Man case] also recognized . . . is that the *Brandenburg* standard was never designed to insulate from liability the communication of detailed information intended to facilitate crime. There was no abstract advocacy in the detailed how-to instructions contained in Hit Man . . . . Hit Man was not political manifesto, not revolutionary diatribe, not propaganda, advocacy, or protest, not an outpouring of conscience or credo. It was *instruction and encouragement*, pure and simple, in the dark arts of mercenary murder.

Smolla, *supra* note 209, at 40–41 (emphasis added).

[220] Some courts have already indicated their reluctance to apply any First Amendment scrutiny to speech that is alleged to have negligently caused harm if the speech at issue does not constitute "advocacy" or fall into some other specific category of "protected" speech. For example, in a wrongful death suit against the Jenny Jones television talk show, where a male participant of the show murdered another male participant who had revealed on the show that he had a crush on the first participant, the trial judge dismissed the defendant's argument that denying the *Brandenburg* instruction could have a chilling effect on talk shows. The trial judge stated that "[i]f that happens so be it, but this court considers this a simple negligence case where people are being called to task for

considered and rejected the application of the *Brandenburg* standard even in cases where the defendant's speech constitutes the sole basis for the claim against the defendant.[221] Civil libertarians will therefore need to base their arguments for the application of heightened intent and causation standards on the particular nature of the speech at issue and the particular nature of the claim brought against the programmer-speaker.

In particular, successful arguments for the application of heightened intent and causation requirements will need to focus on why the speech at issue is "protected" speech.[222] Although it is possible for a speaker to be held liable for "protected" speech, in most cases the classification of speech as "protected" or "unprotected" determines whether or not the speaker will be held liable.[223] The mere fact that the speech at issue is "scientific" or "educational" in the sense that it describes instructions that may be executed by a computer is insufficient to qualify the speech as "protected."[224] Arguments that particular source code constitutes "protected" speech should instead focus on the broader scientific or academic *purpose* of the speech—such as its use in professional research—that goes above and beyond the mere fact that the speech expresses instructions that can be executed by a computer.[225] Only by successfully arguing that the software at issue qualifies as protected

alleged actions which may have done harm to others . . . No more, no less." Goehler & Vollman, *supra* note 168, at 124.

[221] For example, the Georgia Supreme Court applied the *Schenck* formulation of the "clear and present danger" test to the case of a child who was injured when he attempted to reproduce sound effects demonstrated on a television program by rotating a BB inside an inflated balloon. Walt Disney Prod., Inc. v. Shannon, 276 S.E.2d 580, 582 (Ga. 1981). The California Supreme Court declined to apply any First Amendment scrutiny to a claim brought against a radio station for wrongful death based on the death of two motorists who were killed while driving in a metropolitan area in pursuit of a disc jockey who was giving away prize money pursuant to a contest held by the radio station. Weirum v. RKO Gen., Inc., 539 P.2d 36, 48 (Cal. 1975). The court ruled that the First Amendment was not implicated because "[t]he issue here is civil accountability for the foreseeable results of a broadcast which created an undue risk of harm to decedent. The First Amendment does not sanction the infliction of physical injury merely because achieved by word, rather than act." *Id.* at 48. The court in *Rice* stated:

> The First Amendment is quite irrelevant if the intent of the actor and the objective meaning of the words used are so close in time and purpose to a substantive evil as to become part of the ultimate crime itself. In those instances, where speech becomes an integral part of the crime, a First Amendment defense is foreclosed even if the prosecution rests on words alone.

Rice v. Paladin Enters., Inc., 128 F.3d 233, 245 (4th Cir. 1997) (citations omitted).

[222] Failing to demonstrate that the speech at issue is "protected" or constitutes "high-value" speech will likely result in the failure to obtain any First Amendment scrutiny whatsoever. Kagan argues that:

> Courts usually treat the application of a general law, even to activity concededly expressive, as raising no First Amendment issue whatsoever. So, for example, courts will not see a constitutional question if the government convicts for vandalism a person who draws swastikas on a synagogue wall; or applies taxation, labor, or antitrust laws to the publisher of a newspaper; or uses a residential zoning law to prevent the opening of a bookstore.

Kagan, *supra* note 82, at 492.

[223] *See* Wellwood, *supra* note 215, at 189.

[224] *See* Kerr, *supra* note 39, at 1290–92.

[225] *Id.* at 1293.

speech can there be any prospect of courts applying heightened First Amendment scrutiny to claims brought against such speech.

b. Argue Against the Satisfaction of Heightened Requirements in Each Case

Should heightened intent and causation requirements be applied in a particular software speech case, civil libertarians should next focus their arguments on whether the elements of the claims in the case are satisfied. This case-by-case method does not require civil libertarians to forgo the development of more general arguments regarding the applicability of the First Amendment to software speech. It would be worthwhile, for example, to develop general arguments about the beneficial and harmful uses of software and its distribution. Such arguments may be helpful in educating the judge and/or jury about the nature of software and may be tailored to the facts of the case as necessary. These arguments should also embrace the power of software by illustrating how software can be used to perform socially beneficial functions, and how even the distribution of software that performs harmful functions (such as virus software) can be a socially beneficial act. This is because distributing such software in a responsible manner can educate the public about the potential dangers of certain software and enable computer professionals to better understand and develop defenses against it.[226] The success of these arguments will, of course, depend on the facts of the particular case in which they are applied. They are not likely to prevail, and should not prevail, in cases in which the actual defendant distributed software with a malicious intent and in which the actual effect of distributing the software was solely or primarily harmful.

Although this Article focuses only on the elements of intent and causation, other elements and factors will be relevant in particular cases. Other relevant factors may include the likelihood of the harm at issue, the kind of harm, the degree of harm, and the burden that would be imposed on expression by holding the speaker liable.[227]

[226] In Plaintiff's Brief for Felten, it is argued that, "Computer systems research has long been driven by close analysis of existing systems and approaches, in order to understand what works well and what does not. 'It is critical that the researchers and engineers developing new systems be able to study existing ones for advantages and flaws.'" Plaintiffs' Brief in Opposition to RIAA, SMDI and Verance's Motion to Dismiss, at 5, Felten v. RIAA (D.N.J. Aug. 13, 2001) (No. CV-01-2669 (GEB)) (quoting Lazowska Decl. ¶6), *available at* http://www.graysonbarber.com/pdf/brief_opposing_motion_to_dismiss.pdf (last visited Apr. 16, 2004).

[227] For a proposed general approach to the First Amendment as it is implicated in tort cases involving speech by the mass media*, see generally* Sims, *supra* note 93.

C. Non-Legal Solutions Are Also Necessary

Although the preceding discussion focuses entirely on legal solutions to the problems raised by legal regulation of software speech, such solutions are at best a part of a more comprehensive solution to the problem. In cases in which unprotected speech or action contains some kernel of protected expressive content, the law traditionally has relied on the availability of alternative, lawful means for expressing the same message.[228] Although an assassin may not, for example, lawfully express his disagreement with presidential policies by assassinating the President, he may do so by expressing the same message in a pamphlet. In this way, the ability to express the protected message is preserved.

In most cases in which the law prohibits expressing a particular message in a harmful way (such as by assassinating the President), alternative means are practically available to the speaker for expressing the same message in a lawful manner. In the case of software speech, however, prohibiting the distribution of source code may not leave open practical alternative avenues for expressing the same message, because it may not be technologically possible to separate the source code's abstract, scientific, and academic expressive content from the specific instructions contained within the source code for causing a computer to perform unlawful actions.[229] Any attempt to draft a particular regulation

[228] Restrictions on the time, place, and manner of speaking, for example, must "leave open ample alternative channels for communication." Clark v. Cmty. for Creative Non-Violence, 468 U.S. 288, 293 (1984). In the case of most conventional speech, whether a particular regulation leaves open "ample alternative channels for communication" is primarily or solely in the discretion of the regulation's drafter. It is typically within the ability of the drafter, for example, to draft the regulation broadly enough so that "ample alternative channels for communication" are available to speakers. In this sense, the law traditionally has relied on the availability of "ample alternative channels for communication." If it were not possible to draft regulations so that such channels were available, such regulations would never be upheld under the standard just cited.

[229] For example, the court in *Corley* stated that:

> [a]lthough the prohibition on posting prevents the Appellants from conveying to others the speech component of DeCSS, the Appellants have not suggested, much less shown, any technique for barring them from making this instantaneous worldwide distribution of a decryption code that makes a lesser restriction on the code's speech component.

Universal City Studios, Inc. v. Corley, 273 F.3d 429, 454 (2d Cir. 2001). The court phrased this observation as if it were a defect in the appellant's arguments, rather than an effect of the nature of software. However, the court later acknowledged that the nature of software and the Internet:

> obliges courts considering First Amendment claims in the context of the pending case to choose between two unattractive alternatives: either tolerate some impairment of communication in order to permit Congress to prohibit decryption that may lawfully be prevented, or tolerate some decryption in order to avoid some impairment of communication.

Id. at 457–58. *See also* Brief for Amici Curiae EPIC et al., at Part I.B, *Bernstein I*, *supra* note 14 (No. 97-16686) ("[I]t is not merely the case that programming language is expressive; rather, in many instances, such language is the *only* appropriate vehicle for the communication of precise, complex ideas among scientists and mathematicians."); Brief for the Appellee, at Part V.A.2.A., *Bernstein I*, *supra* note 6 (No. 97-16686) ("[B]ecause computer programs are sets of instructions, the government cannot control what those instructions do without also controlling what they say."). Defendant-Appellant's Brief for Corley states:

> Moreover, since computer programs are sets of instructions, restricting the "functionality" of programs also restricts their content. Put more bluntly: if you force a programmer to change what a program does, you inevitably force her to change what it says. Like a law authorizing

sufficiently narrow as to leave open "ample alternative channels for communication" in the form of software speech may result in a regulation so narrow that it no longer covers the harmful activity that it seeks to regulate. The nature of software poses significant problems for a jurisprudence which has relied on relatively bright lines between an idea, a tangible embodiment of the idea (e.g., instructions embodied in source code), and actions which implement the idea.

Technological, normative, and economic solutions will therefore likely be necessary to achieve a desirable degree of protection for expression while effectively regulating harm. Perfect anti-virus software and operating systems that integrally incorporate airtight security measures, for example, would prevent the harm caused by our familiar virus author, thereby making the law moot.[230] Similarly, perfectly and universally internalized ethical norms regarding appropriate means for distributing potentially dangerous software would make the state of technology and the law moot,[231] as would imposing infinite economic costs on culpable virus propagators.[232]

> injunctions to prevent an artist from publishing a work that "functions" to upset audiences while allowing publication of a description of the work or a less upsetting work, the District Court's interpretation of § 1201 requires changes in the content of publications. In computer science, as in art, it is not possible to restrict the form of expression "without also running a substantial risk of suppressing ideas in the process.

Brief for Defendant-Appellant, at Part III, Universal City Studios, Inc. v. Corley, 273 F.3d 429 (2d Cir. 2001) (No. 00-9185), *available at* 2001 WL 34105147 (last visited Apr. 16, 2004) (footnotes and citations omitted).

[230] Increased attention is being paid to the development and implementation of technological measures for securing computer systems, particularly since the terrorist attacks on September 11, 2001. *See generally* Bob Brewin, *Feds Planning Early-Warning System for Internet*, COMPUTERWORLD (Oct. 18, 2002), *available at* http://www.computerworld.com/industrytopics/defense/story/0,10801,75248,00.html; Dan Verton, *Feds Plan Cybersecurity Center*, COMPUTERWORLD (Sept. 2, 2002), *available at* http://www.computerworld.com/governmenttopics/government/policy/story/0,10801,73922,00.html; Steven Levy, *The Big Secret*, NEWSWEEK, June 24, 2002, at 40 (describing Microsoft's proposed security technology named "Palladium"); Wylie Wong, *Shoring Up Software is New Group's Aim*, NEWS.COM (May 16, 2002), *available at* http://news.com.com/2100-1001-915959.html.

[231] Professions, such as the legal profession, have adopted codes of conduct which regulate the speech of members of such professions, just as much as they regulate professionals' conduct. Such regulation is premised, at least in part, on the power that the speech of a lawyer has when addressed to a client. Similar ethical codes may be necessary for computer professionals in response to the potential harm caused by software speech. For an introduction to the significance of computer ethics, see generally James H. Moor, *The Future of Computer Ethics: You Ain't Seen Nothin' Yet!*, 3 ETHICS & INFO. TECH. 89 (2001); *The Ten Commandments of Computer Ethics*, COMPUTER ETHICS INSTITUTE, *at* http://www.cpsr.org/program/ethics/cei.html (last visited Apr. 16, 2004); *Software Engineering Code of Ethics and Professional Practice*, ACM/IEEE-CS JOINT TASK FORCE ON SOFTWARE ENGINEERING ETHICS AND PROFESSIONAL PRACTICES, *at* http://www.acm.org/serving/se/code.htm (last visited Apr. 16, 2004); *ACM Code of Ethics and Professional Conduct,* ASSOCIATION FOR COMPUTING MACHINERY (Nov. 16, 1992), *at* http://www.acm.org/constitution/code.html.

[232] Greater cooperation between software developers/manufacturers and computer security professionals would also be helpful. Some efforts are underway to address the proliferation of defective software using a more cooperative approach than has often been used in the past. *See, e.g.,* Robert Vamosi, *My Plan For Fixing Software Flaws*, ANCHORDESK (Oct. 1, 2002), *at* http://www.zdnet.com/filters/printerfriendly/0,6061,2882094-10,00.html (describing the formation of a committee of five software vendors and security researchers which recommended that: (1) vendors

In practice, of course, it is not possible to achieve any of these ideal goals perfectly and, even if we could, each of them might have unintended negative side effects. In addition, the unique causal powers of software raise new and difficult ethical questions. Our ethical norms governing our responsibility for the consequences of our speech are premised on the limited causal power of various kinds of speech in a wide variety of circumstances. The general rule, for example, that it is neither ethically objectionable nor legally actionable to express a mere wish or desire that the government be overthrown, is premised, at least in part, on the assumption that the expression of such a wish cannot magically cause the government to be overthrown. If, however, we lived in a world in which wish-granting genies existed and roamed the streets freely, and in which uttered wishes were therefore effectively self-executable, each of us would have an ethical obligation not to wish out loud in public for harmful events to occur. This extreme hypothetical is intended to demonstrate that our responsibility for our speech is related, at least in part, to the power of our speech. To the extent that software speech is more powerful than conventional speech, we—and computer professionals in particular—need to think deeply about the ethical implications of such power.

Furthermore, this is not simply a case of fighting to uphold traditional First Amendment principles because, as the discussion above illustrates, the application of traditional First Amendment principles in this case leads to difficult policy questions rather than to a clear outcome. One initially appealing way to make policy decisions would be to prohibit liability for any act of software speech whose expressive content would be protected if expressed in the same circumstances using conventional speech. Such a decision would, however, arbitrarily lock our legal system into rules developed in a time in which we no longer live. We therefore are faced, as always, with the difficult problem of striking the proper balance within and among the legal, technological, ethical, and economic spheres, using a less-than-perfect political process. For this reason, it is even more critical that civil libertarians, in particular, hone their arguments in favor of freedom of expression in the undoubtedly difficult times that lie ahead. If we strike the right balance, we may even promote greater freedom of expression than the somewhat pessimistic legal predictions drawn in this paper indicate.

should acknowledge software vulnerabilities sent by security researchers; (2) vendors should follow up with researchers within ten days; (3) vendors should provide updates every seven days; and (4) vendors should resolve the problem within thirty days).