

¹Code is Speech: Liberalism, Legality, and the Ethics of Free Software

Gabriella Coleman
biella@nyu.edu

Like many computer aficionados today, Seth Schoen, a self-identified hacker, writes all of his software as Free Software so that the source code to his computer programs will remain accessible for other developers and hackers to use, modify, and redistribute.¹ Free and Open Software (F/OSS) developers like Seth not only make technology but in the process of doing so, also reconfigure central tenets in the liberal tradition to produce a rival legal morality to that of intellectual property (IP) law. Take for example an excerpt of a 456 stanza haiku written by Seth in February 2001 that makes a passionate claim that software must be treated as a species of First Amendment speech:

```
Programmers' art as  
that of natural scientists  
is to be precise,  
  
complete in every  
detail of description, not  
leaving things to chance.  
  
Reader, see how yet  
technical communicants  
deserve free speech rights;  
  
see how numbers, rules,  
patterns, languages you don't  
yourself speak yet,  
  
still should in law be  
protected from suppression,  
called valuable speech!
```

¹ While the media usually portrays hackers negatively as information trespassers, here I use the term in a more mundane register to denote computer aficionados who use the term self-reflexively. While the ethical uniformity of hackers is often overemphasized so as to elide the tensions that exist among them, they nevertheless possess a certain concordance in technical and ethical commitments, the nature and effect of which have been the topic of a burgeoning literature (see Kelty 2005, 2008; Nissenbaum 2004; Galloway 2004; Wark 2005; Himanen 2001; Thomas 2002; Levy 1984; Turkle 1984; Sterling 1992).

Seth wrote this poem for a judge overseeing a court case deliberating the First Amendment status of a short piece of Free Software called DeCSS, which could be used to decrypt access controls on DVDs, thus violating current copyright laws. What is notable about the poem is that its author was by no means the only developer at the time to equate source code with expressive speech as a means to challenge intellectual property laws; in fact he was only one among many making similar liberal claims about software that were nearly non-existent in public discourse prior to the 1990s.

Hackers, like Seth, re-configure liberal precepts, such as free speech, through technical practice as well as ideological work. After living and working with hackers in San Francisco and the Netherlands, attending a string of festive hacker conferences, participating in various protests that demanded the release of arrested hackers, and conducting research on the largest Free Software project (Debian), it became clear to me that technologists like Seth—and there are thousands, even tens of thousands of them worldwide—are in the process of transforming long standing liberal principles around property rights, free speech, and the rule of the law. The question is, to what end? And with what political and cultural consequences?

This article examines not only how F/OSS hackers re-articulate liberal principles but demonstrates their political and cultural effects. To do so, I present two contrasting facets of liberal articulations and their consequences, which I encountered simultaneously during my fieldwork. The first, connected to a liberal discourse of political agnosticism, I will describe as a moment of *disavowal* because of the way developers divorce a broad political vision from their labor. The second, a lively series of political protests, I will describe as a moment of *avowal* because of the way hackers and programmers took to the streets between 1999 and 2003 to insist

on their free speech rights to create and circulate software unencumbered by current legal restrictions. While in the first section I focus on political disavowal in order to demonstrate how this political agnosticism facilitates Free Software's broader and powerful political critique of intellectual property law; in the second section I turn to political avowal to examine its cultural consequences, rather than its political ones. During this period of protest F/OSS, hackers enunciated more reflexively than *ever before* their free speech rights to produce and distribute software, which worked to stabilize a relatively new cultural claim whereby source code is imagined as a species of the First Amendment.

The contrasting examples of political disavowal and avowal can be used to demonstrate more richly how the object of democratic, liberal politics might be approached ethnographically as a set of surprising commitments *and* consequences worked out on the ground through the vicissitudes of everyday techno-scientific practice. Several prominent theorists have shown how techno-science, though often configured by a discourse of political neutrality, nonetheless always produces political effects (Mitchell 2002; Ferguson 1994). Yet, they are rarely marked as political, as James Ferguson perceptively argues, precisely because of the way the language of neutrality places techno-scientific projects “under cover of a neutral technical mission to which no one can object” (1994: 273), and thus often work to embolden the power of the state. In contrast, I suggest that it is in fact this technical neutrality which facilitates F/OSS' limited, though vital critique of the corporate and state's regulation of intellectual property law. Further, the tendency for hackers to distance their labor from a political vision outside of software freedom also shapes the types of vocabularies they use during moments of political avowal. Instead of engaging in a broad-based politics, such as demanding justice or equality, or

criticizing capitalism, political expression is configured far more narrowly to secure their productive freedom: the right for developers to freely labor under the conditions of their own making. However, the cultural effects are still profound. During the period of protest discussed below, a new liberal articulation integrally connecting free speech to source code was made more legible through a “structure of the conjuncture,” to use the phrase coined by Marshall Sahlins, through which “categories of being and things” (1981: 72) were endowed with specific and, particular to this case, stable meanings. Thus in what follows, I first examine how hackers distance their labor from traditional politics and the consequences of this disavowal; and then examine how they reacted to state intervention by articulating a novel and rival legal morality exposing the contradictions of liberal intellectual property regimes.

The Politics of Disavowal and the Practical Critique of Intellectual Property Law

A F/OSS developer of the Debian project described the essence of Free Software on a legal mailing list in the following way: “Free Software should create a sort of economy in which things are the way they would be if there were no copyrights at all.” Following this statement, he provides a far more precise sense of how many F/OSS developers conceive of software freedom as something which also demands a form of restraint, neutrality, and political disavowal:

In other words, when I write Free Software, I renounce the ability to control the behavior of the recipient as a condition of their making copies or modifying the software. The most obvious renunciation is that I don't get to demand money for copies. But I also don't get to demand that the person not be a racist; I don't get to demand that the person contribute to the Red Cross. I don't get to demand that the recipient contribute to Free Software. I renounce the little bit of control over the other person which copyright law gives me and in that way, I enhance their freedom. I enhance it to what it would be without copyright law.²

² <http://lists.debian.org/debian-legal/2003/03/msg00494.html>

In other words, to secure the practice of Free Software, this developer claims that one must disassociate the technical practice from other non-technological ideologies, requirements, demands, and affiliations whether they are economic, religious or political.

While this form of disavowal is commonplace in the sphere of F/OSS technical production, this movement has nonetheless taken hold in several, even conflicting social spheres and has worked to politically undermine some central assumptions in intellectual property law. A paradox is at work here: how can a technical movement that often shuns overt political affiliation have initiated such an effective political critique of intellectual property? In what follows, I argue that it is in fact because FOSS is a practice—an act of technical *labor*—that enables it to do so. Because a commitment to the principles of F/OSS are all that binds hackers together and because developers so actively disavow traditional forms of political associations, the broad techno-scientific project of F/OSS has been able to escape the various ideological polarizations (such as liberal vs. conservative) so common in our current political climate. F/OSS has been taken up by a wide array of very differently positioned actors, putting it in a position of significant social legibility whereby it can realize its embedded critique of intellectual property law, even while those who produce it disavow explicit political commitments.

If F/OSS is now a fairly well known technical practice, it once had a far more confined existence. For most of its early life, Free Software was virtually unknown outside of the halls of computer hacking and was largely identified with its intellectual progenitor Richard Stallman, a programmer who came of age at the MIT Artificial Intelligence Lab in the 1970s. His official

quest to thwart the spread of copyrights and patents in software began in 1984 and entailed the creation of the Free Software Foundation, the GNU General Public License, a set of Unix-based tools, and an codified ethical philosophy. One of his most important interventions was a legal one, for the General Public License, more commonly known as the copyleft, provides an alternative to copyright. Copyleft, as its name suggests, inverts the logic of copyright and thus mandates access and openness as well as encourages reuse and modification. By crafting an alternative license, Stallman provided to hackers a means by which to make their socio-technical practice legible before the law; in so doing he also provided the rudiments of a rival liberal legal vocabulary, which hackers would increasingly appropriate, although only years later.

The first major shift in the sociology of Free Software occurred in 1991, with the appearance of the Linux kernel, a software project started by a young Finnish student, Linus Torvalds. While Stallman's actions were intended to guarantee a new era of software freedom, Linus Torvalds' goals were far more modest; yet his actions and the spread of the Linux operating system soon worked to inadvertently confirm the viability of Free Software as a new model of virtual production.

For example, Linux kernel development struck like a tsunami of inspiration and waves of others followed suit. For example, in 1993, Ian Murdock a computer science student emulated the Linux operating system development model to start Debian, a full distribution of the Linux operating system made “by developers and for developers.” During an informal discussion at the annual Debian conference in 2004, Ian explained the idea behind it: “was to get more than one person involved. And the inspiration for that was the Linux kernel. And for some reason the Linux kernel development model seemed to work. . . and I figured, what the hell, let's give it a

try and perhaps we can apply the same idea to this distribution.” As Ian Murdock designed the technical architecture, he, like many other F/OSS project leaders, took it upon himself to conceptualize the nature of this labor. For example, Murdock published the Debian Manifesto, in which he reformulated ideals in Stallman's well-known 1985 GNU Manifesto, to highlight the importance of transparency and distributed collaboration in F/OSS:

The Debian design process is open to ensure that the system is of the highest quality and that it reflects the needs of the user community. . . Involving others also ensures that valuable suggestions for improvement can be incorporated into the distribution during its development; thus, a distribution is created based on the needs and wants of the users rather than the needs and wants of the constructor. . .³

By depicting the architecture of the program as designed not only to promote efficiency but to ensure that the distribution satisfied the “needs and wants of the users rather than [those] of the constructor,” Murdock implicitly asserted a moral, and social, as well as a purely technical end to the project, though one that steered clear of overt political pronouncements.

At the same time as these representations were forming, and in some tension to them, Eric Raymond, a libertarian-leaning hacker, sought to change the ethical terms of engagement in order to attract business investors to F/OSS. To do so, he excised the existing language of freedom and rights by replacing Free Software with an ostensibly more neutral terminology of “Open Source software” and helped pave the way for the integration of Open Source into the corporate sector.

Both terms receive ample usage and usually come paired together, in part, because Free

³ <http://www.debian.org/doc/manuals/project-history/ap-manifesto.en.html>

and Open Source software designate the same licenses and collaborative methodologies. Their differences lie in their linguistic framing (Lakoff 2004) and thus how they formulate the significance of these practices. In F/OSS, we see therefore a plural rather than a singular project: on the one hand, the unmistakably moral position that Free Software projects like Debian adopt; on the other hand, the far more politically neutral position promoted by Raymond and adopted by other projects such as Apache as a *purely* technical tool and method.

Nonetheless, these two sides of F/OSS have more in common than one might think: both share what we might call a 'disavowal of the political,' which is itself of course a form of politics, a configuration made evident to me only when I commenced fieldwork in 2001. At this time, the bifurcation of Free Software and Open Source was firmly in place and led me to believe that among the developers of the Debian project I was working with, I would encounter a political sensibility that exceeded one of software freedom. In fact, Debian, the largest Free Software project with over 1000 developers, and which produces one of the highest quality Linux distributions, is known to have also produced some of the most puritanical Free Software enthusiasts. These technologists have enshrined ethical commitments in an array of artifacts and procedures, including the strictest of Free Software legal guidelines, a Social Contract, Constitution, and complicated membership procedures.

Because of their dense ethical commitment to software freedom, I was surprised to encounter a form of political disavowal. Debian developers routinely police the boundaries of the project so as to prevent certain forms of political associations from entering *official* project policy and even informal discourse. Rather than an absolute distinction between politically engaged Free Software hackers and neutral corporate promoters of the world of Open Source, I

had encountered something more complicated that made me question the well-known distinction between these two positions.

The strongest evidence of this disavowal emerges from what is rarely discussed. Despite the prolific discussions on project mailing lists covering an endless stream of topics—technical problems, project politics, licensing issues, mentoring, project policy—conversations about the role of Debian in supporting widespread political change are nearly non-existent; except, of course, on the rare occasions when someone suggests otherwise.

For example, in the segment here, a developer is disagreeing, vehemently, with another one who suggested in 2003 that Debian should officially participate in a World Social Forum event:

Look, when I signed up for this project and agreed to adhere to the Social Contract, it didn't say anything about Christianity, genetically modified beef, Microsoft, war in Iraq, or anything else like that. It said we agree to work on Free Software. That's the **only** common belief you're guaranteed to find among Debian developers.

Most other developers participating in this incredibly contentious conversation over the project's political scope agreed with his assessment. Many others also pounced on this developer who had dared suggest the existence of a politics beyond software freedom itself.

Certainly some hackers write Free Software to fulfill their own personal political agendas, and more than ever, they simply can't deny the vibrant political life that they themselves have engendered. But as part of their commitments to freedom of expression, liberal individualism, and non-discrimination, many developers divorce a traditional political stance outside of software freedom from official project discourse. Since each has their own personal opinions about politics as well as personal reasons for writing Free Software, hackers believe

these opinions should remain private. This prevents them from attributing a universal political message to their collective labor. This type of sentiment was voiced by many in the email discussion on the World Social Forum, but captured particularly well in the following statement made by a Debian developer:

You must realize that your personal views on other issues are political, and therefore inherently controversial, and are almost certainly not agreed to by every other developer in this project. So let's leave the other politics to the other organizations devoted to them, and keep Debian focused on what it does best.

Here politics is configured as polluting for being personal and thus “controversial” and should be left in the private, not public realm. Pragmatically, the inclusion of politics is deemed undesirable for it may generate unnecessary project strife and interfere with the real task at hand: the production of superior, Free Software, articulated here as what Debian “does best.”

This avowed neutrality is of course a central feature of how liberalism functions as a moral philosophy: by enshrining certain fundamental principles—notably tolerance and free speech—as outside the sphere of the proper domain of politics. Both in liberalism and among F/OSS hackers, these precepts are configured as the apolitical vehicle for securing a just society, individual autonomy, and most significantly, the marketplace of ideas. As Stanley Fish argues, one important idea animating free speech theory is that a “reward” will follow free expression, which “will be the emergence of general and self-evident truths” (2002: 219–20; cf. Peters 2005). By cultivating free expression, hackers also seek to secure a marketplace of ideas that will help establish “self-evident truths.” Yet, as the work of Chris Kelty keenly demonstrates, these self-evident truths are generally limited to what hackers love to obsess over: the functionality, elegance, and worth of technology and, increasingly, the technical means of

association (the Internet) that allows them to collectively associate (Kelty 2005; 2008).

While Debian provides one of the most crystalline examples of how political disavowal emerges through the social structure and discourse of software projects, it is by no means unique. Debian *only stands out* and serves as a useful ethnographic example because it is regarded as the project with the highest ethical standards. Many other software developers, especially Open Source ones, are reluctant to conceptualize their collective labor in ethical terms, much less traditionally political vocabularies. This apolitical stance is also common in the wider hacker cultural sphere (Reimens 2003; Galloway 2004).

Although this disavowal is interesting in its own regard, I've described it to get at one of the most important consequences of the disavowal itself; for although hackers disavow politics among themselves, the consequences of doing so have spilled far beyond this realm of technology to help transform the politics of IP law much more generally.⁴ In adopting and reformulating a liberal politics of expressive freedom and neutrality, F/OSS hackers have effectively coded F/OSS as politically removed, a semiotics of neutrality also significantly magnified by the ambiguity of the term *freedom* in the English language and the contemporary American political climate.⁵ Further, this close symmetry between a value for expressive

⁴ As should be clear by now, I do not seek to reveal the fallacy of liberal neutrality; the critical literature on liberalism has convincingly demonstrated the construction and consequences of such an ordering (Marcuse 1965; Fish 1994; Brown 1995, 2006). Instead, what I am interested in demonstrating is how these erasures acquire meaning within technological social contexts (as opposed to more formal jurisprudence or abstract liberal theory) and what sorts of unexpected consequences they may have in transforming other domains of social life. For similar anthropological approaches to liberalism and neoliberalism see Comaroff and Comaroff (2003), Povinelli (2002), and Ong (2006).

⁵ While my focus here is how F/OSS' free-speech principles help secure F/OSS semiotics of political neutrality, there are of course other important dynamics at play. This form of political disassociation is significantly magnified by the ambiguity of this domain's master trope—freedom—in the English language and the U.S. political context. Of course, as a number of theorists insist: language, words, and especially dialog always tend toward a type of indefiniteness, openness, and multiplicity (Bakhtin 1981; Wittgenstein 1965). However, since the sign of freedom (as well as related ideals of liberal enlightenment, such as “the public” or “science”) rest

freedom and its instantiation in technology—mediated by F/OSS licenses—paves a path for certain socio-political configurations (Winner 1986). Because F/OSS technology is truly free (as in beer as well as in speech)⁶ it is easy for people to use, learn from, and modify—a fact that enables F/OSS to be picked up and translated into a variety of contexts.

For example, a number of hardware companies, most famously IBM, have turned to F/OSS for a service-based business model characteristic of Post-Fordist capitalism. In their advertisements featuring Open Source products and services, they associate it with the positive values of “market agility” and “consumer empowerment” necessary for survival in a flexible workplace. But even while having a foot in corporate America, F/OSS has also been widely adopted by anti-corporate activists, such as those of the Independent Media Centers (IMC). A worldwide and volunteer collective of grassroots media websites and centers, IMC activists make and disseminate locally generated media using various web applications. Many of these tools run on Free Software and activists often conceive of the technology as a radical alternative to the existing corporate-driven software market and products. Meanwhile, a third group of politically-moderate advocates, such as those of the Creative Commons, Public Knowledge, and the student-led movement Free Culture, seek to place some limits on the reach of capitalism by nurturing a space of the “commons,” an idea inspired by F/OSS. But they do not, as the IMC activists do, seek to replace the existing economic system altogether. These three examples are a small fraction of a much wider pool of transfigurations that is growing with each passing day.⁷

atop a trope of universalism, its ambiguity is accentuated, and thus so is its ability to take on various configurations of meaning (Joyce 2003; Warner 2002; Prakash 1999).

⁶ This delineation is a common one by F/OSS advocates, whereby beer means strictly the source code. Even if source code must be made available free, F/OSS is often bought and sold in the open market, but companies usually are charging for service, support, and labor rather than for the knowledge.

⁷ For a detailed discussion of the “modulations” that F/OSS has undergone, see Kelty (2008).

As a result of its appropriation by diverse groups and actors, F/OSS has shifted from relative public obscurity to having a widespread public status. The very practice of Free Software, as a mode of production and a set of licenses, has been legitimated and brought from the subcultural background into the political foreground, as an increasingly unremarkable feature of the contemporary techno-social landscape. It is in part because of its ubiquity today that F/OSS has come to serve as a counter-example to dominant assumptions about the nature of value, labor, freedom, and especially intellectual property in the contemporary era.

If court case after court case, if economist after economist, and all sorts of trade associations stipulate that economic incentives are necessary to induce labor and secure creativity, hackers furnish a counter-stipulation. They do so not simply through the power of rhetoric but through a form of collective labor that is now publicly visible and has yielded high-quality software (software that happens to power much of the Internet). The production of Free Software constitutes a social performance of collective work that contrasts with and effectively chips away at some of the foundational assumptions driving the continual expansion of intellectual property law: namely, the only incentive for production is profit, and that greater freedoms are promoted by guaranteeing intellectual property rights.

To be sure, intellectual property law has never existed free of critique or its own internal contradictions (Hesse 2002; McGill 2002; Johns 2006). However significant these previous critiques were, they never commanded the type of critical influence F/OSS currently does because they did not cohere into a full-fledged movement. F/OSS' now visible collaborative methodologies, ethical philosophy, and most important, alternative licenses have worked as a blueprint to re-imagine *and* reinvent the dominant norms, techniques, and understandings of IP.

In fact, Free Software licenses such as the GPL and similar copyleft licenses rupture the naturalized form of intellectual property, inverting its singular logic through the use of intellectual property itself. Recall that F/OSS licenses are built upon copyright, but they disable restrictions by adding a clause that allows for and encourages access and copying. This move is not unlike Marx's inversion of Hegelian idealism, which retained Hegel's dialectical method to re-posit history not as an expression of the "Absolute Idea" but as humanity's collective creation through labor. Using copyright as its vehicle, the copyleft turns copyright on its head and in the process demystifies copyright's "absolute" theory of economic incentive. Just as significant, it opens to question the idea that only "institutions of authority," notably the U.S. courts and Congress, can determine the rule of law.

Thus, its most profound political effect has been to devitalize the hegemonic status of IP law via a practical politics of visibility, opening it to reflection and active contestation. The moment that "any set of values, and material forms comes to be explicitly negotiable," as John and Jean Comaroff remind us, marks the end of its naturalized state (1992: 29). In emphasizing this practical and critical dimension of F/OSS, I am echoing a common theme within a rich literature in anthropology, sociology, and cultural studies (cf. Comaroff 1985; Taussig 1987; Ong 1987; Hebdidge 1979; Gilroy 1993). This literature compellingly argues that the scope of politics far exceeds intentional acts and encompasses non-rhetorical or aesthetic dimensions that in *some* historical instances can speak louder than any directed rhetorical program. But the question always remains: how can such a critical politics actualize itself in a fuller sense by becoming public, captivating an audience, offering critical insight, and moving others to join?

The answer to this question requires us to assess the relationship between a dominant

political climate and the pragmatic and semiotic elements specific to a phenomenon under investigation. In the prevailing political climate, where mere identification with the Right or the Left, conservatives or liberals, can work as form of ideological imprisonment, F/OSS has been able to successfully avoid such ghettoization to gain presence across various sectors of society.⁸ In *this* particular historical moment, F/OSS' political ambiguity and replicable nature facilitate its ability to captivate a diverse audience who are provoked to action because they have confronted a living piece of evidence, one that they are willing to entertain because it has not been ostensibly "tainted" by an easily identifiable political program. It behaves as what Bruno Latour calls "a theater of proof" (1988) and thus has become a model for organizing similar endeavors.

To be sure, the debate over the direction, scope, and limits of intellectual property law is certainly not over, just because we can point to an existing alternative that has been at the basis of catalyzing similar endeavors. In some respects because of the forceful appearance of F/OSS, a formidable politics *has just begun*, fueled by the rise of technologies like peer-to-peer systems that facilitate copying, translation, and reconfigurations. Those who abide by dominant principles under threat may clamor for more attention. For example, on the eve of an important Supreme Court deliberation on the legality of peer-to-peer technologies, the *New York Times* ran an editorial stating its position on IP law. The editors chose a vocabulary of gloom and doom: "If

⁸ In the United States, since the 1980s, the political polarization between conservatives and liberals and Republicans and Democrats has not only been pronounced but also accompanied by increasing suspicion and denigration of "the left" or "liberals." And indeed, most attempts to contain F/OSS within an explicit political framework have been made by Microsoft or conservative think-tanks seeking to portray F/OSS either as excessively liberal, or even worse in the American mainstream imaginary, socialist or communist. For a discussion of the conditions that secured current forms of political polarization, see Abramowitz and Saunders (1998). For a series of *New York Times* articles on the devaluation of the term liberal and liberalism in the 1980s, see Dionne (1988), Haufler (1988) and Lipset (1988).

their work is suddenly made ‘free,’ all of society is likely to suffer. . . . The founders wrote copyright protections into the Constitution because they believed that they were necessary for progress.”⁹ By invoking the founding fathers and tropes of progress, this message sought to reassert the naturalness of certain liberal propositions precisely when they were most under threat. This recent pronouncement was matched by a series of anti–Open Source reports and articles churned out by think tanks bearing names like Alexis de Tocqueville Institution and The Progress Freedom Foundation. Such names speak to the fact that we are in the midst of a pitched political (and cultural) battle over the proper meaning and scope of liberal freedom and intellectual property law.

And yet, even as the editors of the *New York Times* decried the corrosive consequences of a more liberal IP regime, the hackers themselves steadily maintained an ideology of political neutrality. While F/OSS developers were quite critical of the direction of IP law, their projects did not align themselves with other political movements, such as those connected to the World Social Forum and social justice movements that might have seemed so natural an ally. Nonetheless, F/OSS still functioned politically. By virtue of the fact that one can point to a living practice that contradicts arguments based on abstract principles, the latter tend to lose some of their efficacy. On this basis, policy and law can perhaps be more easily channeled away from universal claims and can entertain local, pragmatic stakes and address *the reality of what people do, can do, or desire to do*. As Helen Nissenbaum has argued, we are drawn to the example of hacker activity because hackers “represent a degree of freedom, an escape hatch from a system that threatens to become overbearing” (2004: 212). Since hackers and their

⁹ “When David Steals Goliath's Music.” <http://www.nytimes.com/2005/03/28/opinion/28mon1.html> March 28, 2005.

artifacts are usually either in legally dubious waters or on the cusp of new legal meaning, their ethical pronouncements and practices, such as those of Free Software, reveal much broader emerging or contentious dilemmas over the fate of law and technology. Rendered visible, many scholars and lawyers have already taken the example of F/OSS to argue powerfully for balancing the current system and they do so in the midst of other activists and educators who shore up their own arguments not within the pages of books but by building alternatives (Bollier 2002; Lessig 1999; Benkler 2006).

Code is Free Speech: The Politics of Protest and Avowal

Even if political disavowal is common within Free Software projects, we can also readily locate many instances of a more traditional politics of struggle among F/OSS developers, also stepped in liberal idioms. Here I turn to one such instance: a set of lively protests launched against new intellectual property regulations (notably the DMCA), and the arrest of two programmers, Jon Johansen and Dmitry Sklyarov, which led to some of the most widespread hacker protests of the last decade. Hackers took these arrests as the most virulent example of an intellectual property system run amok—as a system they felt was a threat against *all* F/OSS hackers and their productive freedom.

These protests reveal a different side of democratic politics on the ground in the context of techno-science; one similarly connected to the re-articulation of liberal ideals but in this case, wedded to a politics of avowal. If political agnosticism contributed to a practice-based and non-rhetorical critique of IP, then these protests, which I analyze under the rubric of “critical events,” launched an overt and rhetorical critique of new intellectual property regulations. The consequence of these protests and critical events was to engender a more durable and public

articulation of a free-speech ethic among Free Software developers than ever before.

Thus, what I highlight here is not so much the political effects of these protests, but instead their notable cultural consequences. I was drawn to tell the story of these protests because I witnessed, and even in some cases, participated in a series of events through which hackers constructed a more definitive claim about the expressive nature of source code than I had ever encountered. Following theorists of narrative such as Paul Ricoeur, we can see how critical events help fix nascent cultural mores because of the ease by which participating actors can set such events (and thus themselves) within a narrative arc. As part of a process of “*emplotment*,” defined by Ricoeur as “a synthesis of heterogeneous elements” (1991: 21), hackers stitched their particular local experiences in F/OSS production with those of the arrests. During this process, they established a narrative that stabilized a new association between source code and free speech.¹⁰ The outcome of these critical events was a form of social cathexis—an *intensification* of cultural processes already under way, but now staged more publicly as protests and cultural artifacts that clearly expressed, and in some cases, artistically performed the types of connections between source code and free speech that hackers had only *very recently* started to fashion into a coherent set of discourses.

The story of these critical events began on October 6, 1999 with a seemingly insignificant act. A sixteen-year-old Norwegian programmer, Jon Johansen, released a short software program, DeCSS, on a mailing list. Co-authored by himself and two anonymous

¹⁰ For Ricoeur, a narrative takes shape when there is a temporal ordering of disparate events that are invested with enough meaning so as to develop into a coherent storyline (1985, 1991). William Sewell's recent discussion also adds crucial specificity to our theoretical understanding of how critical events can secure social reproduction and transformation (2005). My own treatment is indebted to his typology of ten characteristics and many of these, which he formulates by drawing on the French Revolution—such as heightened emotional tone, collective creativity, a proliferation of events—are pertinent to the example discussed here.

developers, DeCSS is a simple piece of software: it works to unlock access controls, more commonly known as Digital Rights Management (DRM) on DVDs. At the time of release, DVDs could only be played on Microsoft's Windows and Apple's operating system and Johansen's program allowed Linux users to unlock a DVD's DRM in order to play movies on their computers. Released under a Free Software license, it soon made its way across the globe available for download on hundreds, possibly thousands, of websites. In the hacker public, the circulation of DeCSS would also transform Johansen from an unknown geek into a famous "freedom fighter;" entertainment industry executives, on the other hand, would soon seek out his arrest.

While many geeks were gleefully using this technology to bypass a form of Digital Rights Management so they could watch DVDs on their Linux machines, various trade associations sought to ban this piece of software that also made it easier to copy and thus pirate DVDs.¹¹ In November 1999, soon after its initial spread, the DVD Copy Control Association and the MPAA (Motion Picture Association of America) sent cease-and-desist letters to more than fifty website owners and Internet service providers, requiring them to remove links to the DeCSS code for its alleged violation of trade secret and copyright law and, in the United States, the Digital Millennium Copyright Act (DMCA). In January 2000, the MPAA filed three lawsuits against different web publishers for publishing DeCSS. One was served against one of the most well-known hacker organizations and publications named *2600*, and its founder Eric Corley (more commonly known by his hacker handle, Emmanuel Goldstein). He would soon fight the

¹¹ Despite the fact that one could use DeCSS to pirate DVDs, there was no compelling evidence that upon its widespread adoption it was being used for the types of large-scale pirating efforts that were already common at the time.

courts on various appeals on the grounds that *2600* had a journalistic free-speech right to publish DeCSS. As happens with censored material, the DeCSS code at this time was unstoppable; it spread like wildfire.

Since the DMCA figures prominently in both arrests and lawsuits addressed here, I shall briefly summarize this statute, explain why the MPAA resorted to it, and why it has been met with fierce resistance among hackers and other First Amendment loyalists. Passed in 1998 with the aim of “modernizing” copyright for the digital age, its most controversial provision outlaws the manufacture and trafficking of technological tools (which can mean something immaterial, such as a six-line piece of source code, or something physical in nature) capable of circumventing copy or access protection measures used in copyrighted works that are in a digital format. It is vital to underscore that the DMCA makes illegal the trafficking and circulation of such a tool, even if it can be used for lawful purposes (such as fair use copying) or is never used.¹²

While these lawsuits were unfolding, the international arm of the MPAA contacted Norwegian authorities about Johansen’s involvement with DeCSS and urged prosecution under Norwegian law (the DMCA, an American law, had no jurisdiction there). The Norwegian Economic and Environmental Crime Unit (ØKOKRIM) took the MPAA’s informal legal advice and indicted Johansen on January 25, 2000, for violating an obscure Norwegian criminal code. Johansen (and his father) were arrested and released on the same day, and law enforcement confiscated his computers. He was scheduled to face trial three years later.

¹² Because of this troubling preemptive feature, a handful of prominent scholars and lawyers critique this provision as draconian, for it obliterates the already fragile fair-use doctrine explicitly underpinning copyright law since 1976 (Lessig 2001; Samuelson 1999; Vaidhyathan 2001, 2004; Litman 2001; Gillespie 2007).

Hackers and other geek enthusiasts discussed, debated, and decried these events on every conceivable forum. Like many Internet discussions, the responses were so prolific and so varied that it is difficult to condense them into a pithy summary. However, a few consistent themes and topics soon emerged. Among the first was the marked influence of another court case concerning software and speech, *Bernstein v. United States*; this case established that software could be protected under the First Amendment. The case opened in 1995, after a computer science student, Daniel J. Bernstein, sued the government to challenge international traffic in arms regulations (ITAR), which classified certain types of strong encryption as munitions and thus subjected them to strict export controls. Because of these provisions, Bernstein could not legally publish or export the source code of his encryption system, Snuffle, without registering as an arms dealer. After years in court, in 1999 the judge presiding over the case concluded that government regulations of cryptographic “software and related devices and technology are in violation of the First Amendment on the grounds of prior restraint.”¹³

It is clear that many F/OSS advocates thought of the DeCSS cases in light of the Bernstein case and its victory also fueled the hope that the courts just might declare DeCSS worthy of First Amendment protection. For example, the first message posted on the mailing list “dvd-discuss”—a mailing list that would soon attract a small army of programmers, F/OSS developers, and activist lawyers to discuss every imaginable detail concerning the DeCSS cases, including the technical and ontological relationship between source code and expression—flags the importance of the Bernstein case:

Greetings,

¹³ http://www.eff.org/Privacy/Crypto_export/Bernstein_case/19990507_eff_pressrel.html

This is my first post to dvd-discuss, so I will kick things off by introducing myself [. . .]
I see the DVD cases as the natural complement to Bernstein's case. Just as free speech protects the right to communicate results about encryption, so it protects the right to discuss the technicalities of decryption. In this case as well as Bernstein's, the government's policy is to promote insecurity to achieve security. This oxymoronic belief is deeply troubling, and worse endangers the very interests it seeks to protect.¹⁴

However, upon closer inspection there were significant differences between the Bernstein case and the DeCSS ones. Given the slow pace of legal developments with the Bernstein case (the case was decided four years after it was first filed) and *especially* its inability to induce a string of other non-legal responses, hackers were positioned primarily as engaged spectators, viewing the case as a bounded event. Further, a number of Free Software advocates were quite critical of Bernstein's decision to copyright, and thus tightly control, all of his software. In the contrast, with the DeCSS and DVD cases, many F/OSS hackers quickly integrated themselves as participants and did so by injecting a language of Free Software, free speech and source code into the fabric of the unfolding cascade of events, which they in fact helped to precipitate.

Indeed, many of the initial comments following Johansen's indictment clearly reflect how hackers also conceptualized his arrest and the lawsuits as a violation of their right to produce F/OSS, propelling many into direct action. As the following passionate call to arms reveals, many hackers understood the regulation of DeCSS as a “full-fledged war against the Open Source movement:”

... here's why they're doing it: **Scare tactic**. They want to "get tough" on the "pirates" and scare people into submission. . . I know a lot of us aren't political enough - but consider donating a few bucks and also mirroring

¹⁴ <http://web.archive.org/web/20031124051048/cyber.law.harvard.edu/archive/dvd-discuss/msg00000.html>

the source . . . This is a full-fledged war now against the Open Source movement: they're trying to stop reverse-engineering and black-box everything. They can justify and rationalize all they want - but it's really about them trying to gain/maintain their monopoly on distribution. . .¹⁵

Hackers viewed Johansen as an unfair target of a law that challenged, in fundamental ways, the freedom to tinker and to write code without restrictions—a set of values that hackers had only come to collectively rationalize with more depth and breadth in the sphere of F/OSS production in the previous years.

Soon after Johansen's arrest and the lawsuits, many hackers were inspired to fight back and swiftly moved to organize politically. A vast number of websites providing highly detailed information about the DMCA, DeCSS, and copyright history went live, and the Electronic Frontier Foundation (EFF) launched a formal "Free Jon Johansen" campaign.

Within this flurry of political mobilization, another trend soon emerged: the call to re-circulate DeCSS as an act of civil disobedience. Again, the rhetoric accompanying the call tended to be rooted in the language of Free Software, as this post by Evan Prodromou, a Debian Developer and editor of one of the first Internet zines, *Pigdog* reveals:

Hey, so, I've been [really mad](#) about the recent spate of horrible witch hunts by the [MPAA](#) against people who use, distribute, or even LINK TO sites that distribute DeCSS, a piece of software used for playing DVDs on Linux. The MPAA has got a bee in their bonnet about this DeCSS. They think it's good for COPYING DVDs, which, in fact, it's totally useless for. But they're suing everybody ANYWAYS, the bastardos! Go look at [OpenDVD.org](#) to find out lots more about this really despicable travesty.

Anyways, I feel like I need to do something. I've been talking about the the whole travesty here on Pigdog Journal, of course, and helped with the

¹⁵ <http://slashdot.org/comments.pl?sid=3644&cid=1340340>

[big flier campaign](#) here in SF and everything, but I feel like I should do something more, like help redistribute the DeCSS software.

There are a lot of problems with this, obviously. First and foremost, Pigdog Journal is a collaborative effort, and I don't want to bring down the legal shitstorm on the rest of the Pigdoggers just because I'm a Free Software fanatic. Also, I'm a total chicken, and don't want to go to court

DeCSS is Born

So, I decided that if I couldn't distribute DeCSS, I would distribute DeCSS. Like, I could distribute another piece of software called DeCSS, that is perfectly legal in every way, and would be difficult for even the DVD-CCA's lawyers to find fault with. [. . .]

Distribute DeCSS!

I encourage you to distribute DeCSS on your Web site, if you have one. The newest version, 0.07, includes an out-of-the-box mirroring kit that should make it easy as pie. I think of this as kind of an "I am Spartacus" type thing. If lots of people distribute DeCSS on their Web sites, on Usenet newsgroups, by email, or whatever, it'll provide a convenient layer of fog over the OTHER DeCSS. I figure if we waste just FIVE MINUTES of some DVD-CCA Web flunkey's time looking for DeCSS, we've done some small service for The Cause.¹⁶

Concurrent with these types of responses and political actions, the DeCSS lawsuits were unfolding and these proved crucial in stabilizing the growing links between source code and software, largely because they generated the most comprehensive statements to date that computer code is expressive speech, which in turn, were read, discussed, and nearly unanimously supported by F/OSS developers. Particularly prominent among these was the *amicus curiae* brief on the expressive nature of source code written by a prominent group of computer scientists and hackers, as well as the testimony by one of its authors, Carnegie Mellon Computer Science professor, David Touretzky, a fierce and well-known free speech activist. For example, *Linux Weekly News*, a widely circulated newsletter with the "latest news from the Linux World," provided the following overview and subsequent analysis of Touretzky's

¹⁶ <http://www.pigdog.org/decss/>

testimony:

The 2600 case (wherein [2600 Magazine](#) is being sued by the MPAA in the form of Universal City Studios for having mirrored the DeCSS code) has concluded testimony in New York. Some of the final developments in the trial include:

[. . .]

- Final-day testimony from David Touretzky, who talked about the free speech implications of the DeCSS injunction. His point was that the restriction of source is equivalent to a restriction on speech, and would make it very hard for everybody who works with computers.

The judge responded very well to Mr. Touretzky's testimony, saying things like "*I was hoping we were going to hear something like this through the whole trial.*" and "*I think one thing probably has changed with respect to the constitutional analysis, and that is that subject to thinking about it some more, I really find what Professor Touretzky had to say today extremely persuasive and educational about computer code.*"

[. . .]

Thus, there are two rights being argued here. One is that of reverse engineering - we have the right to look at things we own and figure out how they work. We even have the right to make other things that work in the same way. The other is that code is speech, that there is no way to distinguish between the two. In the U.S., of course, equating code and speech is important, because protections on speech are (still, so far) relatively strong. If code is speech, then we are in our rights to post it.

If these rights are lost, Free Software is in deep trouble. . .¹⁷

In this exegesis we see again how Free Software developers framed the importance of these cases by interweaving Free Software, source code, and free speech. These connections I want to reemphasize were *absent* in hacker public discourse prior to the early 1990s. While Richard Stallman certainly grounded the politics of software in a liberal vocabulary of freedom; and Daniel Bernstein's fight introduced a far more legally sophisticated idea of the First Amendment for software; it would only be after the DeCSS cases when a more prolific and specific language

¹⁷ <http://lwn.net/2000/0727/bigpage.php3>

of free speech would come to dominate the ethical discourse among F/OSS developers.¹⁸ If the law, to use Clifford Geertz's famous formulation is “part of a distinctive manner of imagining the real” (1981: 184), these types of responses to DeCSS and the lawsuits demonstrate how a legal precept—free speech—became culturally real. In this period, hackers securely fastened the idea of free speech to a technical artifact, source code, and in a much broader capacity than ever before.

So far I have highlighted the importance of the prolific set of discursive and legal responses in constituting the connections between source code and speech by providing just a handful of exemplary responses from a much wider pool. The most significant response, however, was neither purely discursive nor legal but emerged in what should be seen as a culturally-inflected form of protest: the playful and clever re-creations of the original DeCSS source code (originally written in the C programming language) into other languages (such as Perl) and, perhaps more surprisingly, into other media such as poetry, music, and film. Collectively compiled on the website hosted by David Touretzky, “The Gallery of CSS

¹⁸ My aim here is not to provide a standard history of the rise of expressive rights among programmers and hackers, but to examine the conditions under which a more nascent association between source code and speech was secured. However, since no comprehensive history exists, here I provide some background information. The first widely circulated paper associating free speech and source code was “Freedom of Speech in Software” (1991) written by a programmer, Peter Salin. He characterized computer programs as “writings” to argue that software was unfit for patents although appropriate for copyrights and free-speech protections. The idea that coding was a variant of writing (and thus deserved free-speech protections) held sway among hackers, because by this time there was growing recognition of the literary nature of programming (Knuth 1998; Black 2002). Meanwhile, what have come to be known as the “encryption wars” were waged for the right to freely publish and use cryptography in the face of governmental restrictions that classified strong forms of encryption as munitions and the most notable juridical case in these struggles was *Bernstein v. U.S. Department of Justice*. However, while Salin's influential article and Bernstein's case crucially contributed to the establishment of a language and politics seeking First Amendment recognition for software, neither questioned the validity of copyright law as a barrier against speech. What Free Software philosophy added to this history was a *more fundamental* critique of the idea of property in challenging patents and copyrights and, much more important, in cultivating a set of social practices and licensing schemes that provided a rival legal morality, steeped in an idiom of free speech, to the dominant regime of IP law.

DeScramblers,”¹⁹ it showcased a set of twenty-four artifacts whose point was to demonstrate the difficulty of drawing a sharp line between functionality and expression in software. It thus, performed the intellectual arguments about free speech and code being posed by the defendants in the *2600* trials and informally by developers on all sorts of mailing lists.²⁰ David Touretzkey made this message clear on the Gallery website with the following introductory statement:

If code that can be directly compiled and executed may be suppressed under the DMCA, as Judge Kaplan asserts in his preliminary ruling, but a textual description of the same algorithm may not be suppressed, then where exactly should the line be drawn? This web site was created to explore this issue, and point out the absurdity of Judge Kaplan's position that source code can be legally differentiated from other forms of written expression.

Along with conveying an unmistakable political message, these transmutations of DeCSS are potent cultural artifacts: taken together, they illustrate how hackers have come to conceptualize source code as speech and provide a glimpse of how they conceive of creativity in hacking.

Before I present two refashioned examples and their cultural significance, here is a short snippet, about one quarter in length, of the original DeCSS source code written in the C programming language:

```
void CSSdescramble(unsigned char *sec,unsigned char *key)
{
    unsigned int t1,t2,t3,t4,t5,t6;
    unsigned char *end=sec+0x800;

    t1=key[0]^sec[0x54]|0x100;
    t2=key[1]^sec[0x55];
```

¹⁹ <http://www-2.cs.cmu.edu/~dst/DeCSS/Gallery/>

²⁰ I am bracketing one of the most interesting questions raised here: is source code speech and how do we draw the line or imagine the relationship between functionality and expression in source code? These were some of the most important questions that programmers and hackers as well as the court cases struggled to answer during this period. The discussions on these topics were incredibly varied but since most were quite lengthy and technical, it is not feasible for me to portray them here except to point out that most programmers insisted on the expressive nature of source code, comparing computer languages to the written word or natural languages. For a legal analysis of the relationship between code and speech, see Burk (2000) and Tien (2000). Ratto (2005) provides a thoughtful response to these dominant legal paradigms, which tend to dichotomize expression and functionality in code. Instead, he analyzes expression in code within a broader framework that includes their material affordances and their social production.

```

        t3=((unsigned int *) (key+2))^((unsigned int
*) (sec+0x56));
        t4=t3&7;
        t3=t3*2+8-t4;
        sec+=0x80;
        t5=0;
        while(sec!=end)
        {
            t4=CSStab2[t2]^CSStab3[t1];
            t2=t1>>1;
            t1=((t1&1)<<8)^t4;
            t4=CSStab5[t4];
            t6((((t3>>3)^t3)>>1)^t3)>>8^t3)>>5)&0xff;
            t3=(t3<<8)|t6;
            t6=CSStab4[t6];
            t5+=t6+t4;
            *sec+=CSStab1[*sec]^(t5&0xff);
            t5>>=8;
        }
    }

void CSStitlekey1(unsigned char *key,unsigned char *im)
{
    unsigned int t1,t2,t3,t4,t5,t6;
    unsigned char k[5];
    int i;

    t1=im[0]|0x100;
    t2=im[1];
    t3=((unsigned int *) (im+2));
    t4=t3&7;
    t3=t3*2+8-t4;
    t5=0;
    for(i=0;i<5;i++)
    {
        t4=CSStab2[t2]^CSStab3[t1];
        t2=t1>>1;
        t1=((t1&1)<<8)^t4;
        t4=CSStab4[t4]; [...]
    }
}

```

Compare this fragment to another one written in Perl, a computer language that hackers regard as particularly well suited for crafting poetic expressions because longer expressions can be condensed into much terser, sometimes quite elegant (though sometimes quite obfuscated) statements. And indeed the original DeCSS program, composed of 9,830 characters, with Perl was condensed significantly into a mere 530 characters:

```

#!/usr/bin/perl -w

# 531-byte qrpff-fast, Keith Winstein and Marc Horowitz
# <sipb-iap-dvd@mit.edu>
# MPEG 2 PS VOB file on stdin -> descrambled output on stdout
# arguments: title key bytes in least to most-significant order

```

```

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

```

If Perl allows programmers to write code more “poetically” than other computer languages, Seth Schoen took up the challenge of publishing a bona fide poem in the form of an epic haiku—456 individual stanzas written over the course of just a few days. An abstract yet accurate representation of DeCSS, the poem was written to deliver a political message to Judge Kaplan who was presiding over the *2600* case. The message is stark and clear. The author asserts that source code is not a metaphor or “similar to expression” but *is* expression and he makes this point by recreating the original DeCSS program within what is typically seen as the most creative of literary genres, the poem. This bit of poetry was both a cultural artifact of high value and a protest. Because such combinations are rare in hackerdom, it is a luminous object for analysis.

Its author, who at the time of publication remained anonymous because of potential legal threat, chose this genre in order to appeal to specific aesthetic sensibilities. Schoen explains: “A strange tradition current among programmers calls for the use of the 5-7-5 pattern—preferably cleverly—to express technology, or jokes about technology, or really anything at all, just for the fun or the challenge of writing within the constraint” (2004). Indeed, creativity is not viewed as the production of absolute or pure novelty. Instead, hackers play with all sorts of formal constraints—from hardware specifications to inherited code to the 5-7-5 pattern of a haiku—to create what they collectively designate as clever, innovative, or elegant solutions. In other

words, hackers routinely use a system, piece of code, or artifact for more than what these were invented for; within constraint, they seek to innovate.

The author opens his poem first by thanking Professor Touretzky and then moves immediately to abandon his “exclusive rights” clause of the copyright statute, indexing the direct influence of F/OSS licensing:

How to Decrypt a DVD: in haiku form

(Thanks, Prof. D. S. T.)

(I abandon my
exclusive rights to make or
perform copies of

this work, U. S. Code
Title Seventeen, section
One Hundred and Six.)

Muse! When we learned to
count, little did we know all
the things we could do

some day by shuffling
those numbers: Pythagoras
said “All is number”

long before he saw
computers and their effects,
or what they could do

by computation,
naive and mechanical
fast arithmetic.

It changed the world, it
changed our consciousness and lives
to have such fast math

available to
us and anyone who cared
to learn programming.

Now help me, Muse, for
I wish to tell a piece of
controversial math,

for which the lawyers
of DVD CCA
don't forbear to sue:

that they alone should

know or have the right to teach
these skills and these rules.

(Do they understand
the content, or is it just
the effects they see?)

And all mathematics
is full of stories (just read
Eric Temple Bell);

and CSS is
no exception to this rule.
Sing, Muse, decryption

once secret, as all
knowledge, once unknown: how to
decrypt DVDs.

Here, the author first frames the value of programming in terms of mathematics, which goes unrecognized by a trans-local sphere of power. This author identifies this “power” in the entertainment industry, IP statutes, lawyers, and judges, all of which and whom are seen to signify and value the nature of software production in particular and different ways. In fact, he claims they exploit and feed off the products of programming without recognizing, much less truly understanding, the creative labor and expressive value embedded in software. The writer indicates this difference of signification over value in the form of a question: “Do they understand the content, or is it just the effects they see?”

The writer then launches into a *very* long mathematical description of the forbidden CSS code represented in DeCSS. The portion provided below represents only a minuscule portion of the technical segment. This example outlines the “player key” of CSS, which is the proprietary piece that enacts the access control measures:

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

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

Two hundred and twenty
four; and last (of course not least)
the humble zero

Because he represents the access control mathematically in words, from these lines alone a proficient enough programmer can deduce the encryption key—and thus this portion of the poem makes a similar point to the one made in the *amicus* brief, namely that “[a]t root, computer code is nothing more than text, which, like any other text, is a form of speech. The Court may not know the meaning of the Visual BASIC or Perl texts . . . but the Court can recognize that the code is text.”²¹

After the technical portion and claiming that software is mathematical, the poet conveys how many F/OSS software programmers, like himself, have also come to conceive of their craft as a technically precise (and thus functional) yet fundamentally an *expressive* activity rooted in the practice of writing and as a result worthy of free speech protection. In formally comparing code to poetry, and this is important, in the medium of a poem, he is displaying a playful form of recursive rhetoric valued among hackers (Fischer 1999) and uses it as a springboard by which to re-valorize both the meaning of the First Amendment and software among a public beyond his engineering and hacker community:

We write precisely
since such is our habit in
talking to machines;

we say exactly
how to do a thing or how
every detail works.

The poet has choice
of words and order, symbols,
imagery, and use

²¹ <http://www.2600.org/dvd/docs/2001/0126-speech.html>

of metaphor. She
can allude, suggest, permit
ambiguities.

She need not say just
what she means, for readers can
always interpret.

Poets too, despite
their famous "license" sometimes
are constrained by rules:

How often have we
heard that some strange twist of plot
or phrase was simply

"Metri causa", for
the meter's sake, solely done
"to fit the meter"?

This haiku, thus, uses culturally recognizable aesthetics to politically defend a technical way of life considered to be under threat through the indiscriminate application of new forms of intellectual property law. Even though this dramatic enunciation, on the one hand, clearly reflects the existence of novel assertions (in this case the tight coupling between source code and speech), it is also through its inscription into a tangible and especially a culturally captivating medium (a hack with playful and recursive qualities) by which these types of assertions are transformed into firm social fact. What we see here is how public discourse, as Michael Warner has noted, "helps to make a world insofar as the object of address is brought into being partly by postulating and characterizing it" (2002: 91).

Taken together: the legal strategizing on mailing lists and forums; the websites defending Jon Johansen; the recirculation of DeCSS as well as the refashioned versions; and protests all demonstrate how a pool of programmers and hackers quickly became active participants who weaved themselves into the unfolding drama concerning law and Free Software in the digital age. These protests on their own enabled what legal theorist Robert Cover describes as a simultaneous process of subjective commitment to and objective projection of norms, a bridging

that emerges out of a narrative mode. “This objectification of the norms to which one is committed frequently,” Cover writes, “perhaps always entails a narrative—a story of how the law, now object, came to be, and more importantly, how it came to be one’s own” (1992: 145).

This narrative process described by Cover, however, would only continue to grow in importance in the midst of the DeCSS cases and protests due to another, unrelated DMCA infraction and the arrest of another programmer, Dmitry Sklyarov. Because Sklyarov was arrested for violating the DMCA and faced up to 25 years in jail, programmers in fact only grew more infuriated with the state's willingness to police technological innovation and software distribution through this statute. After Sklyarov's arrest, the politics of avowal and protest against the DMCA and the hacker commitment to a discourse of free speech only grew in visibility and emotional intensity and worked to extend the narrative process already well underway.

In comparison to Johansen's arrest, this one would also prove far more dramatic simply due to a contingent reason: the timing and place of his arrest. Dmitry Sklyarov was arrested while leaving Defcon, the largest hacker conference in the world held annually in Las Vegas. During the conference, Sklyarov had presented a paper on security breaches and weaknesses within the Adobe e-book format. He purportedly had violated the DMCA for his role in making a piece of software for his Russian employer, Elcomsoft, which works to unlock Adobe's e-book access controls and subsequently convert it into the more common PDF format. For the FBI to arrest a programmer at the end of this conference was a potent, unmistakable statement. It testified to the fact that federal authorities would act on corporate demands to prosecute hackers under the DMCA. To clarify this claim, let me briefly describe the particularities of the great ritual event of the hacker conference, commonly known in shorthand as the “con.”

During Defcon, hackers spend four days denying their bodies' basic biological needs— notably sleep—so they can hack, party, and play with friends they usually interact with at length but largely only on-line. It is an intense, effervescent, and thoroughly ritual affair held yearly in America's strangest vacation playground, Las Vegas. Serious technical talks go hand-in-hand with parties, dancing, swimming, gambling, and an endless stream of games, events, and activities. They range from a three-day nonstop "hacking" tournament in which teams attempt to crack an encrypted server, to "hey lets go check out Area 51 again," the secret military base of several shadowy "alien experiments."

FBI agents always attend this conference, but there is a well-known, although tacit, agreement that these federal agents, immediately identifiable by their L.L. Bean[®] khaki attire in a sea of black clothing, T-shirts, and body piercings, do not interfere with the hackers. Considered by most to be somewhat annoying but not seriously harmful, agents have been playfully integrated into Defcon's carnivalesque spirit by being targeted in games, for example, "Spot the Fed."²² Despite their presence since the 'con began in 1993, FBI agents had *never* arrested a hacker at this conference (local police authorities, of course, have made multiple arrests, but usually based on charges of excessively rowdy and drunken behavior). The first ever FBI arrest of a hacker at Defcon sent a strong signal that intellectual property infractions were now serious criminal and federal offenses, resulting in a re-negotiation of the relationship between legal authority and the hacker world.

On July 17, 2001, as Sklyarov was leaving the conference, federal agents whisked him

²² In "Spot the Fed," an audience member can interrupt a formal technical presentation to point out the presence of an agent; such interruptions are encouraged. The agent's presence is announced loudly—sometimes klaxons are sounded—and the agent is nudged to the front of the room to undergo a playful ritual of public humiliation, an expression of the tensions between law enforcement and the hacker underground that is tolerated because of its humorous veneer.

away to an undisclosed jail in Nevada. Weeks later, he was released in the middle of a fervent “Free Dmitry” campaign. Sklyarov's arrest and related court hearings prompted the continued prolific and pronounced set of discussions and responses that had been initiated by Jon Johansen's arrest and the resultant DeCSS lawsuits. However, the Johansen protest efforts paled in relation to the Free Dmitry campaign, which was organized more swiftly, was more visible, and directly attacked Adobe, the company that had urged the Department of Justice to make the arrest.

At that time, I was undertaking my fieldwork in San Francisco, which became a central hub of political mobilization. Protests and other activities were attended and organized by many local F/OSS developers, even though Sklyarov was in *no way* affiliated or identified with the world of F/OSS development. Protests were staged across American cities (Boston, New York, Chicago, San Francisco, among others), in Europe, as well as Russia. In the San Francisco Bay Area, rallies and events included a protest at Adobe’s San Jose headquarters, a candlelight vigil at the San Jose public library, and a march held after Linux World on August 29, 2001, that ended up at the federal prosecutor’s office (see Figure 1).

Figure 1. Free Dmitry Protest at the Federal Prosecutors Office, San Francisco, CA.

After the march to the prosecutor's office, where protesters affirmed and enunciated the connection between speech and code (evidenced in the photo above), activists hosted a well-attended fund raiser. There, Richard Stallman, the founder of the Free Software Foundation, and Lawrence Lessig, the superstar activist-lawyer who held some sway in the hacker community, gave impassioned speeches to hundreds of programmers and hackers after a short guest appearance by Dmitry Sklyarov himself, who thanked the audience for their support. The mood was electric in an otherwise cool San Francisco warehouse loft. Lessig, who had recently published a book that was changing the way F/OSS developers understood the politics of technology, *Code and Other Laws of Cyberspace*, fired up the already animated crowd with charged declarations:

Now this is America, right? It makes me sick to think this is where we are. It makes me sick. Let them fight their battles in Congress. These million-dollar lobbyists, let them persuade Congressmen about the sanctity of intellectual property and all that bullshit. Let them have their battles, but why lock this guy up for twenty-five years?

Most programmers agreed with Lessig's assessment: the state had gone way too far in its uncritical support of the copyright industries.

These varied expressions of opposition were in some respects immediately effective. After only five days of a "Ban Adobe" campaign and especially the embarrassing protests at Adobe's headquarters, the company withdrew its support of the case. Eventually, the court dropped all charges against Sklyarov on the condition that he testify in the case subsequently

directed against his employers, which he did. In December 2002, the jury in that case, much to the chagrin of the presiding judge, acquitted Elcomsoft. A little over a year later, in January 2003, Johansen was also acquitted for the case was seen as far too shaky for prosecution, as the law he was arrested under had nothing to do with digital rights management. Johansen still writes Free Software (including other software that subverts DRM technologies), as well as a blog cleverly called “So Sue Me,”²³ and has become a de facto hero among F/OSS hackers.

The DeCSS lawsuits were also decided in the course of 2001–2004 with varied outcomes. Because of its violation of copyright law, which U.S. courts currently and consistently judge to be the proper conduit for the marketplace of ideas, the government felt it had reason to suppress source code as speech. Even though every court in these cases agreed that DeCSS was speech, they deemed DeCSS as only deserving an “intermediate level of scrutiny,” and thus courts could override the First Amendment arguments made by the defendants. In one of the 2600 cases, *Universal City Studios Inc. v. Reimerdes*, Judge Kaplan went so far as to declare that the court's decision meant to “contribute to a climate of appropriate respect for intellectual property rights in an age in which the excitement of ready access to untold quantities of information has blurred in some minds the fact that taking what is not yours and not freely offered to you is stealing.”²⁴

²³ While the name “So Sue Me” is an obvious reference to Johansen's legal woes, the name—like many hacker names—has another, in this case more obscure, meaning. Those who are well versed in the history of computer technology and law will immediately recognize that “So Sue Me” is also a reference to “sosumi,” one of the system sounds in Apple’s Macintosh System 7 operating system, a name born out of another legal dispute. Due to ongoing trademark disputes between the Beatles' home label, Apple Records, and Apple Inc. Computers, Apple Inc. was barred from including musical-sounding names. All names had to be approved by the Apple Inc. legal team. The author of this system’s sounds, Jim Reekes, had grown frustrated with all the legal red tape and came up with “sosumi”—a replacement for the original name, “chimes”—as a clever critique and, more pragmatically, as a way to bypass the scrutiny of the legal team, as he claimed the word was a Japanese one having nothing to do with sounds.

²⁴ *Universal City Studios Inc. v. Reimerdes*, 82 F. Supplement 2d 211 [2000]. This case was appealed by one of

After the decision made by Judge Kaplan, the author of DeCSS haiku, Seth Schoen published a response, which would also circulate widely on the Internet, “The History of the DeCSS Haiku.” He not only provides the back story to the poem, but includes a piercing cultural analysis of the outcome of the DeCSS cases. He again directly questions Judge Kaplan's assumptions on the nature of the First Amendment and reaffirms how and why developers like himself were re-working liberal precepts:

Yet ultimately much comes down to cultural framing of a free speech controversy *as* a free speech controversy: if a court accepts that something is “a First Amendment case”, the speaker is extremely likely to win, especially if the party on the other side is a government. It's hard to avoid the inherent *sympathy* Judge Patel bears toward Professor Bernstein (a speaker whose expression is crushed by the awesome might of government bureaucracy) or the equally apparent suspicion with which Judge Kaplan regards Emmanuel Goldstein (a self-avowed hacker seemingly hell-bent on trouble). These attitudes seem to me to be visible behind all the doctrinal questions; . . . I would say that Judge Patel fought to show why her case was a free speech case and that Judge Kaplan fought to show why his was not. The question of which approach seems natural would then be not primarily a question of legal doctrines, standards, or precedents. It would instead be a conceptual, cultural battle: shall programs be compared to epidemics of disease (evil, menacing, worthy only of quarantine) or to books in libraries (the cornerstones of our culture and our civilization)? (Schoen, 2004)

Even if the courts refused to protect DeCSS under the First Amendment, which left many F/OSS developers like Seth Schoen deeply disappointed, his response above nonetheless reveals how the First Amendment, as most legal precepts, is also a matter of “cultural framing” and thus has a rich life in the form of everyday legal consciousness (Silbey and Ewick 1998; Mezey 2003). The law is as much an everyday cultural reality as it is a matter of legal doctrine and these

the defendants, Eric Corley. In the subsequent case, *Universal City Studios Inc. v. Corley* 273 F. Supplement 3d 429 [2001], the presiding judges also affirmed the importance of this view in so far as they highlighted and quoted a longer version of this statement.

arrests and cases in fact helped to secure it as cultural commonplace within the hacking community. With narratives like these now so firmly in place, hackers, programmers, and computer scientists will be motivated in the near future to transform *what is now their* cultural reality—a rival liberal morality—into a broader legal one by launching and supporting lawsuits which argue that source code is and should be protectable speech under the U.S. Constitution.

Conclusion: “This is What Democracy Looks Like”

During the period under which the Jon Johansen and Dmitry Sklyarov protests occurred, there was also a blizzard of more well-known protests in the United States and Europe, which were fueled by the politics of anti-corporate globalization and whose dramatic public face was showcased with the WTO protests in Seattle in 1999, the same year DeCSS was released. One of the most famous rallying calls of the anti-corporate globalization movement was, “This is What Democracy Looks Like!” In certain respects, the two contrasting expressions of political avowal and disavowal I examined here, could not be more different from the types of actions and discourses enunciated by these activists. While these anti-corporate activists were attacking a neoliberal world order and denouncing corporate abuse in the name of greater social justice and economic equity, most F/OSS developers were either simply concerned with producing and securing Free Software, or defending their productive autonomy. Hackers are often bound together by an unquenchable dedication to technology that makes them ambivalent about directing and conceptualizing their praxis towards anything but itself. As Patrice Reimens has incisively argued, this makes hackers less likely to make “demands for justice, equality, emancipation, [and] empowerment” (2003) so common within traditional political programs and radical political movements.

However, once we take a more expansive and ethnographic view of what democracy on the ground looks like and within the context of techno-science, it is clear these technologists are intimately engaged in various types of democratic struggles by sustaining a rival liberal legal morality, one which has catalyzed important political and cultural transformations. Even if F/OSS developers as a collective rarely express anti-corporate or anti-capitalist sentiments, this domain has still contributed to one of the most powerful critiques ever waged of intellectual property law—one important regime helping to sustain a neoliberal capitalist economy (Harvey 2005).

F/OSS' lack of political affiliation and portability are two conditions that facilitate its politics of visibility and critique. The contemporary political climate of ideological polarization was another key factor in a convergence that has allowed the labor of Free Software to perform an embedded and practical critique and throw into question cherished assumptions driving the theory of incentive that has dominated views of creativity and invention for well over two hundred years.

At the same time, this domain has been the site for a more traditional and rhetorically-based politics of protest also meant to halt the application and spread of intellectual property instruments, notably the DMCA. The unrelated and contingent arrests of Jon Johansen and Dmitry Sklyarov, in particular, proved a greater boon to the consolidation of the anti-DMCA movement than to the suppression of so-called piracy, one of the motivations to ratify the DMCA in Congress at the turn of the century. As significant is how these protests performed crucial cultural work. In the course of these events, relatively new connections between source code and speech came to be intimately associated with each other. Through a cumulative process

that continued across several years of lively protests, ongoing responses, and prolific discussions, the connection between source code and speech was naturalized into the commonsense state under which it now exists today.

Works Cited

Abramowitz, Alan I. and Kyle Saunders

1998 Ideological Realignment in the U.S. Electorate. *Journal of Politics*. 60: 634-65.
39(3):323-349.

Bakhtin, Mikhail

1981 *The Dialogic Imagination*. Caryl Emerson and Michael Holquist, trans. Austin: University of Texas Press.

Benkler, Yochai

2006 *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. New Haven: Yale University Press.

Bollier, David

2002 *Silent Theft: The Private Plunder of Our Common Wealth*. New York: Routledge.

Black, Maurice J.

2002 *The Art of Code*. PhD dissertation, Department of English, University of Pennsylvania.

Boyle, James

1996 *Shamans, Software, and Spleens*. MA: Harvard University Press.

2003 *The Second Enclosure Movement and the Construction of the Public Domain*. *Law and Contemporary Problems* 66: 33-74.

Brown, Wendy

1995 *States of Injury: Power and Freedom*. Princeton: Princeton University Press.

2006 *Regulating Aversion: Tolerance in the Age of Identity and Empire*. Princeton: Princeton University Press.

Burk, Dan

- 2000 Patenting Speech. 79 Texas Law Review 99.
- Comaroff, Jean
- 1985 Body of Power Spirit of Resistance. Chicago: University of Chicago Press.
- Comaroff, Jean and John Commaroff
- 1992 Ethnography and the Historical Imagination. Boulder: Westview Press.
- 2003 Reflections on Liberalism, Policulturalism, and ID-ology: Citizenship and Difference in South Africa. *Social Identities* 9(3): 445-74.
- Cover, Robert
- 1992 Nomos and Narrative. *In Narrative, Violence, and the Law: The Essays of Robert Cover*. Martha Minow, Michael Ryan, and Austin Sarat, eds. Pp. 95-172. Ann Arbor: The University of Michigan Press.
- Dionne Jr., E. J.
- 1988 Describing Liberalism. *New York Times*, November 1: A27.
- Editorial
- 2005 When David Steal's Goliath's Music. Op-ed Editorial. *New York Times*.
Electronic document, <http://www.nytimes.com/2005/03/28/opinion/28mon1.html?ex=1269666000&en=a1ff3d9ace6d8734&ei=5088&partner=rssnyt>.
- Ewick, Patricia and Susan Silbey
- 1998 The Common Place of the Law. Chicago: University of Chicago Press.
- Fish, Stanley
- 1994 There's No Such Thing As Free Speech: And It's a Good Thing, Too. Oxford, England: Oxford University Press.
- 2002 The Dance of Theory. *In Eternally Vigilant*. Lee C. Bollinger and Geoffrey Stone, eds. Pp. 198-231. Chicago: University of Chicago Press.
- Fisher, Michael
- 1999 "Worlding Cyberspace: Towards a Crucial Ethnography in Time, Space, Theory." *In Critical Anthropology Now: Unexpected Context, Shifting Constituencies, Changing Agendas*. George Marcus ed. Santa Fe: Sar Press.
- Ferguson, James
- 2006 [1994] The Anti-Politics Machine. *In The Anthropology of the State*. Aradhana Sharma and Akhil Gupta, eds. Pp. 270-286. Oxford: Blackwell.
- Galloway, Alexander R.
- 2004 Protocol: How Control Exists after Decentralization. Cambridge, MA: The MIT

Press.

Geertz, Clifford

1981 *Local Knowledge: Further Essays in Interpretive Anthropology*. New York: Basic Books.

Gillespie, Tarleton

2007 *Wired Shut: Copyright and the Shape of Digital Culture*. Cambridge: The MIT Press.

Gilroy, Paul

1993 *The Black Atlantic: Modernity and Double Consciousness*. Cambridge: Harvard University Press.

Harvey, David

2005 *A Brief History of Neoliberalism*. Oxford: Oxford University Press.

Haufler, Hevie

1988 *Rekindling the Good Spirit Behind the 'L' Word*. *New York Times*, December 4: CN42.

Hebdige, Dick

1997 [1979] *Subculture the Meaning of Style*. In *The Subcultures Reader*. Ken Kelder and Sarah Thorton, eds. Pp. 13-142. New York and London: Routledge (393-405).

Hesse, Carla

2002 *The Rise of Intellectual Property, 700 B.C.-A.D. 2000: An Idea in the Balance*. *Daedalus* (Spring): 6-45.

Himanen, Pekka

2001 *The Hacker Ethic and the Spirit of the Information Age*. New York: Random House.

Johns, Adrian

2006 *Intellectual property and the nature of science*. *Cultural Studies* 20 (2&3): 145-64.

Joyce, Patrick

2003 *The Rule of Freedom: Liberalism and the Modern City*. London and New York: Verso.

Kelty, Chris

2005 *Geeks, Social Imaginaries, and Recursive Publics*. *Cultural Anthropology* (20)2:

185-214.

2008 Two Bits: The Cultural Significance of Free Software and the Internet. Durham: Duke University Press.

Knuth, Donald

1998 The Art of Computer Programming (vol 1-3). New York: Addison-Wesley.

Lakoff, George

2004 Don't Think of an Elephant: Know Your Values and Frame the Debate--The Essential Guide for Progressives. White River Junction, Vermont: Chelsea Green.

Latour, Bruno

1988 The Pasteurization of France. Catherine Porter, trans. Cambridge: Harvard University of Press.

Lipset, Seymour Martin

1988 Americans Sneer at Liberalism. Why? New York Times, October 28: A1.

Litman, Jessica

2001 Digital copyright: Protecting Intellectual Property on the Internet. Amherst, NY: Prometheus Books.

Lessig, Lawrence

1999 Code and Other Laws of Cyberspace. New York: Perseus Books.

2001 The Future of Ideas: The Fate of the Commons in a Connected World. New York: Random House.

Levy, Steven

1984 Hackers: Heroes of the Computer Revolution. New York: Delta.

Marcuse, Herbert

1965 Repressive Tolerance. *In A Critique of Pure Tolerance*. Pp. 81-123 . Boston: Beacon Press.

McGill, Meredith

2002 American Literature and the Culture of Reprinting, 1834-1853. Philadelphia: University of Pennsylvania Press.

Mezey, Naomi

2001 Law as Culture, 13 Yale Journal of Law and Humanities.

Mitchell, Timothy

- 2002 *Rule of Experts: Egypt, Techno-Politics, Modernity*. Berkeley: University of California Press.
- Nissenbaum, Helen
- 2004 Hackers and the Contested Ontology of Cyberspace. *New Media and Society* (6)2:195-217.
- Ong, Aihwa
- 1987 *Spirits of Resistance and Capitalist Discipline: Factory Women in Malaysia*. Albany: State University of New York Press.
- 2006 *Neoliberalism as Exception: Mutations in Citizenship and Sovereignty*. Durham: Duke University Press.
- Peters, John Durhman
- 2005 *Courting the Abyss: Free Speech and the Liberal Tradition*. Chicago: University of Chicago Press.
- Prakash, Gyan
- 1999 *Another Reason: Science and the Imagination of Modern India*. Princeton: Princeton University Press.
- Povinelli, Elizabeth
- 2002 *The Cunning of Recognition: Indigenous Alterities and the Making of Australian Multiculturalism*. Durham: Duke University Press.
- 2006 *The Empire of Love: Toward a Theory of Intimacy, Genealogy, and Carnality*. Durham: Duke University Press.
- Ratto, Matt
- 2005 *Embedded Technical Expression: Code and the Leveraging of Functionality, The Information Society*. vol. 21(3).
- Ricoeur, Paul
- 1985 *Time and Narrative*, vol. 3. Kathleen Blamey and David Pessauer, trans. London and Chicago: University of Chicago Press.
- 1991 *Life in Quest of Narrative*. *In* *On Paul Ricoeur; Narrative and Interpretation*. David Wood, ed. Pp. 20 – 33. New York and London: Routledge.
- Riemens, Patrice
- 2003 [2002] Some Thoughts on the Idea of 'Hacker Culture'" *Multitudes* 8 (2).
Electronic document, <http://multitudes.samizdat.net/Some-thoughts-on-the-idea-of.html>, accessed April 2, 2004.
- Sahlins, Marshall

- 1981 Historical Metaphors and Mythical Realities: Early History of the Sandwich Islands Kingdom. Ann Arbor: University of Michigan Press
- Salin, Peter
- 1991 Freedom of Speech in Software. Electronic document, <http://www.philsalin.com/patents.html>, accessed November 12, 2003.
- Samuelson, Pamela
- 1999 Intellectual Property and the Digital Economy: Why the Anti-Circumvention Regulations Need to Be Revised. Berkeley Tech L. J. 14: 519- 548.
- Sewell, William
- 2005 Logics of History: Social Theory and Social Transformation. Chicago: University of Chicago Press.
- Schoen, Seth
- 2001 How to decrypt a DVD: in haiku form. Electronic document, <http://www.cs.cmu.edu/~dst/DeCSS/Gallery/decss-haiku.txt>, accessed October 2, 2002.
- 2004 The History of the DeCSS Haiku. Electronic document, <http://www.loyalty.org/~schoen/haiku.html>, accessed January 10, 2005.
- Sterling, Bruce
- 1992 The Hacker Crackdown: Law and Disorder on the Electronic Frontier. New York: Bantam.
- Taussig, Michael
- 1987 Shamanism, Colonialism, and the Wild Man: A Study in Terror and Healing. Chicago: University of Chicago Press.
- Thomas, Douglas
- 2002 Hacker Culture. Minneapolis: University of Minnesota Press.
- Tien, Lee
- 2000 Publishing Software as a Speech Act. Berkeley Technology Law Journal 15(2): 629-712.
- Turkle, Sherry
- 1984 The Second Self: Computers and the Human Spirit. New York: Simon and Schuster.
- Vaidyanathan, Siva
- 2001 Copyrights and Copywrongs: The Rise of Intellectual Property and How it

Threatens Creativity. New York: New York University Press.

2004 The Anarchist in the Library: How the Clash between Freedom and Control is Hacking the Real World and Crashing the System. New York: Basic Books.

Warner, Michael

2002 Publics and Counterpublics. New York: Zone Books.

Wark, McKenzie

2004 A Hacker Manifesto. Cambridge, MA: MA: Harvard University Press.

Winner, Langdon

1986 The Whale and the Reactor: A Search for Limits in an Age of High Technology. Chicago: University of Chicago Press.

Wittgenstein, Ludwig

2001 [1953] Philosophical Investigations. Third edition. Gertrude Elizabeth Margaret, trans. Oxford: Blackwell.