Chicago-Kent Law Review

Volume 84
Issue 3 *Symposium: Data Devolution: Corporate Information Security, Consumers, and the Future of Regulation*

Article 13

June 2009

Coding Privacy

Lilian Edwards

CODING PRIVACY

LILIAN EDWARDS*

INTRODUCTION

Regulation, in Larry Lessig's famous insight, comes not just from laws, but also from norms, from the market, and from the architecture of the world. This is true in cyberspace and in real space. Take, for example, the very topical issue of the regulation of the copying of digital works. We know that these works can be protected by civil copyright law, but they can also be protected—or at least the RIAA has tried to protect them[1]—by the propagation of the norm that file sharing is actually stealing. As another example, every time a film or movie is released, it can be protected through various means in the market, for example, by simultaneous rather than staggered releases in different geographical markets. But crucially as a digital object, it can also be protected by the architecture of the cyberspace world. That is by using "code" such as digital rights management (DRM).

When I use the term "code," I mean not just program code (i.e. lines of code C++ or the like), but also the physical environment of cyberspace or, in other words, the entire architecture composed of the software, the hardware, the wires, the devices attached by wires, and even "offline world" devices attached by wireless internet, which include RIFD chips,[2] for example.

I. ARCHITECTURES

In *Code and Other Laws of Cyberspace*, Larry Lessig demonstrates how cyberspace is governed by the interaction of the four regulating forces

* Lilian Edwards is Professor of Internet Law at the University of Sheffield. She can be reached at lilian.edwards@sheffield.ac.uk.

[1] *See, e.g.*, Recording Industry Association of America, RIAA Pre-Lawsuit Letters Go To 22 Campuses In New Wave of Deterrence Program, http://www.riaa.com/newsitem.php?id=8E8AE31D-2CD9-5E90-8892-5FEBD3A603B9 (last visited Dec. 6, 2009). For a discussion of the recording industry and their lawsuits relating to digital music filesharing, see, for example, Ryan Bates, Comment, *Communication Breakdown: The Recording Industry's Pursuit of the Individual Music User, a Comparison of U.S. and E.U. Copyright Protections for Internet Music File Sharing*, 25 NW. J. INT'L L. & BUS. 229 (2004).

[2] For more information on RFID chips and their uses, see Scott Granneman, *RFID Chips Are Here*, SECURITY FOCUS, June 26, 2003, http://www.securityfocus.com/columnists/169.

I have identified—architecture, law, market, and norms.[3] Let us start with Lessig's famous statement that, in cyberspace, architecture or "code" regulates like legal code.[4] In the real world, architecture is a given. For example, we do not need legislation forbidding humans from flying in built up areas because gravity regulates that already, and gravity is a naturally occurring architecture in the real world. Cyberspace is different in that the architecture has been built from scratch by deliberate design decisions. Someone has designed it, someone wrote it, and someone owns it. That is the crucial point, despite it being fairly obvious. What is less obvious is the idea that this "code" is not designed with the constraints that are normally applied to legal codes, such as attention to the impact on fundamental human rights.

A. The Democratic Deficit

Code is not made by elected democratic representatives, and it is not subject to constitutional control. It is not subject, for example, to human rights control, such as the European Convention on Human Rights that we are familiar with in the UK and the EU. It is not subject to any of the Constitutional Amendments. So, we have what I call the "democratic deficit" of code, which can lead us to the idea that rather than living in a world regulated by elected governments, we might be living online in the "Microsoft world" or even the "open-source world."

Although many "real world" laws apply, of course, to cyberspace activities, the meat of this insight is that cyberspace is also regulated by coders and not exclusively, or even primarily, by law. What Lessig does not quite say—though you can argue that it is there between the lines—is that code is superior to law as a means of effective regulation.[5] I however will say it: code (often, though not always) trumps law.

B. How Code Trumps Law

What I find in my own research increasingly is that whatever legal problems or technology problems our laws attempt to regulate, code is more effective at achieving the aim of the regulation. Take again the example of digital copying. While the law has been almost completely ineffec-

[3] *See* LAWRENCE LESSIG, CODE AND OTHER LAWS OF CYBERSPACE (ver. 2.0, 2006). For additional discussion of these distinctions, see Lawrence Lessig, *The New Chicago School*, 27 J. LEGAL STUD. 661, 662–63 (1998).

[4] *See* LESSIG, CODE, *supra* note 3, at 5.

[5] *See id.* at 120–37.

tive in preventing digital copying,[6] Digital Rights Management (DRM) code is very effective.[7] An arms race may of course develop of code vs. code—as has been the case with DRM algorithms and widely publicized technical hacks developed to circumvent them. More recently, DRM has fallen out of favour due to a consumer backlash against it, but a new "code" solution—disconnecting the Internet accounts of alleged downloaders at the request of the music industry—has become popular with lobbyists for the content industry and adopted into the laws of both the UK (the Digital economy Act of 2010) and France ("HADOPI").

Similarly, there has been a large amount of anti-spam laws passed at the state and federal levels[8] in the U.S., and there have been EC Directives and local laws passed on privacy, data protection and spam control in the E.U.[9] None of these laws have had a meaningful effect on preventing the flow and dissemination of spam and the percentage of email that is spam continues to rise.[10] However, Spam filters—a "code" solution—make my inbox a happy place. I am no longer troubled by spam (though, it has to be said, my ISP still is) because I have Spam Assassin,[11] and that solves the

[6] *See* Zachary Williams, Note, *Hometaping In the Twenty-First Century: Updating the Audio Home Recording Act to Address Emerging Technologies*, 36 AIPLA Q.J. 77, 81–82 (2008).

[7] Digital rights management is defined as a code-based method for authorizing the viewing or playback of copyrighted material on a user owned machine. *See, e.g.*, ZDNet, DRM: Definition, http://dictionary.zdnet.com/definition/DRM.html (last visited Dec. 27, 2009). For a discussion of digital rights management, see, for example, Stefan Bechtold, *Digital Rights Management in the United States and Europe*, 52 AM. J. COMP. L. 323 (2004) (arguing statutory limitations to the different means of DRM protection seem necessary); Dan L. Burk, *Legal and Technical Standards in Digital Rights Management Technologies*, 74 FORDHAM L. REV. 537 (2005) (examining the social costs of deploying digital rights management systems to protect copyrighted content); Julie E. Cohen, *DRM and Privacy*, 18 BERKELEY TECH. L.J. 575 (2003) (arguing that with some adjustments, DRM technologies could be harnessed to protect privacy).

[8] The primary statute in the United States addressing spam is the CAN-SPAM Act, Controlling the Assault of Non-Solicited Pornography And Marketing, 15 U.S.C. § 7701 (2006). For a discussion of CAN-SPAM, see, for example, Scot M. Graydon, *Much Ado About SPAM: Unsolicited Advertising, the Internet, and You*, 32 ST. MARY'S L.J. 77 (2000) (arguing federal legislation is needed to harmonize state level spam regulation); Elizabeth Phillips Marsh, *Purveyors of Hate on the Internet: Are We Ready for Hate Spam?*, 17 GA. ST. U. L. REV. 379 (2000) (arguing that in the absence of governmental regulation of privacy data, criminal law should be able to reach hate spam if it encourages violence to an intolerable degree); Gary Miller, *How to Can Spam: Legislating Unsolicited Commercial E-Mail*, 2 VAND. J. ENT. L. & PRAC. 127 (2000) (arguing that advertisers should be required to obtain permission before they send advertisements through an Internet service provider's system).

[9] For discussion of EU approaches to spam, see, generally, LILIAN EDWARDS, LAW AND THE INTERNET, ch. 15 (Edwards, L. & Waelde C., eds., 3d. ed. 2010).

[10] *See* Taiwo A. Oriola, *Regulating Unsolicited Commercial Electronic Mail in the United States and the European Union: Challenges and Prospects*, 7 TUL. J. TECH. & INTELL. PROP. 113 (2005); Jeffrey D. Sullivan & Michael B. De Leeuw, *Spam After CAN-SPAM: How Inconsistent Thinking Has Made a Hash Out of Unsolicited Commercial Email Policy*, 20 SANTA CLARA COMPUTER & HIGH TECH. L.J. 887 (2004).

[11] Spam Assassin is an open source spam filter. Spam Assassin, http://spamassassin.apache.org/ (last visited Jan. 27, 2009).

problem for me. So, again, code trumps law.

II. CODE AND PRIVACY

If you go back to look at the original version of *Code and Other Laws of Cyberspace*, the privacy chapter[12] was quite evasive and very pessimistic. It implied that code had to be inherently invasive of privacy.[13] Lessig gave examples such as surveillance by cameras, websites we visit, and data-collection companies and the subsequent mining of the data collected.[14] Lessig's preferred solution to the anti-privacy regulatory bias of code at the time was a "privacy enhancing technology" (PET), namely P3P, the platform for privacy preferences.[15] The underlying idea of P3P was that users would basically go out into the electronic world and bargain at arms' length about what data they wanted to give away and at what price they were willing to do so for every data-collecting website.[16] The trouble with P3P was that consumers, lacking education or intuition about the risks of disseminating their personal data, had no incentive to spend this time on bargaining and even more importantly, the market had little or no incentive to pay or negotiate for data that they had previously collected for free. The model though, simply did not succeed. Although P3P was incorporated into Internet Explorer and other browsers,[17] it has been largely ignored by the public and the market. No meaningful marketplace of choices among more or less privacy friendly websites evolved for the consumer. So Lessig picked the wrong solution. But does this mean that code in cyberspace always has to be inherently anti-privacy? This writer would disagree with that assessment.

My approach to code, law, and privacy starts with the idea of technology neutrality, that is, the presumption that code is not necessarily invasive of privacy. It is not inevitable that any increase in the amount of code on the Internet will lead to an accompanying decline in the amount of privacy on the Internet, even though this is the impression you might sometimes derive from looking at the website of the EFF,[18] EPIC,[19] the Berkman Cen-

[12] LESSIG, CODE, *supra* note 3, at 200–32.

[13] *See id.* at 202–03.

[14] *Id.* at 203–04, 215–16.

[15] *Id.* at 228.

[16] *Id.* at 226.

[17] *See, e.g.*, Microsoft Corporation, Use Internet Explorer 6 to Help Safeguard Your Privacy on the Web, http://www.microsoft.com/windows/ie/ie6/using/howto/privacy/config.mspx (last visited Jan. 27, 2009).

[18] Electronic Frontier Foundation, http://www.eff.org/ (last visited Dec. 27, 2009).

[19] Electronic Privacy Information Center, http://epic.org/ (last visited Dec. 27, 2009).

ter,[20] Privacy International and FIPR, and the like. However the correct incentives have to be in place for the owners and writers of code to prioritize privacy. Merely bolting on a specific privacy-enhancing technology tool—like P3P—after the fact, to a system based around largely privacy-invasive code and anti-privacy values is unlikely to work. A better solution is to design code to promote privacy as a core value to start with—the idea of "privacy by design." The question then is how do we make corporations that write and install code consider privacy as a plus and not a minus in their software development cycle *to start with*?

The first insight is that code, like other technologies, is to begin with, a value-neutral tool. It is, however, then imbued with the values of those who own it and those who write it. These are not necessarily the same persons. A seemingly infinite number of coders work for Microsoft, but the values imbued into the code are almost certainly connected to what makes profits for Microsoft—or perhaps Bill Gates' ideas of Microsoft's core business mission—rather than these coders' personal ethics. Thus if Microsoft makes money from collecting data about its users, then, barring regulation, it has little incentive to design its code to be privacy-enhancing. This is pretty much the status quo. I discuss below how to make privacy look like part of a business model and less like a problem for a business dependent on data collection.

It is worth noting that some argue that since code can be rewritten—modified or hacked—the values of the original "owners" are not crucial and code always remains "value–neutral." However the average consumer or user—people like my mother—does not know how to code. Average consumers do not know how to encrypt their email, for example, or their P2P traffic.[21] They do not know how to protect themselves against private or public electronic surveillance or how to "kill" RFID chips by putting them in a microwave. These people do not have the power generally to tweak code to their own end. Thus, the values of the original writers and packagers of code remain crucial.

A. Anti-Privacy and Pro-Privacy Code

In this section, we will look at some examples of pro and anti-privacy code, drawn from the EU experience, and then in the next section, try to work through what incentivized the use of privacy-enhancing or privacy-

[20] Berkman Center for Internet & Society, http://cyber.law.harvard.edu/ (last visited Dec. 27, 2009).

[21] *See, e.g.*, Bittorent, http://www.bittorrent.com/ (last visited Dec. 27, 2009).

invasive code.

"Code," in the metaphorical sense I am using, occurs in the offline and online world. In the offline world, a good example is closed-circuit television cameras (CCTV) which are ubiquitous in the UK.[22] When I walk out of my office, there are cameras in the halls, cameras outside the doors, cameras in the street, cameras in front of stores, cameras at the airport, and so forth. Ostensibly, these cameras are there for preventing or detecting crime, and this enables them to ignore the terms of almost all of the controls required under UK data protection (DP) law.

Another example of anti-privacy code in the offline world relates to RFID tags. We are seeing increasing use of RFID in the commercial private sector as inventory aids and anti-shop lifting devices,[23] and they are also increasingly used to track and monitor users via the likes of biometric passports, school and library books and transport smartcards.[24] RFID chips allow recording of what users do in the real world and especially tracking *where* they go, such as what transport routes they take. This "bugging" quality facilitates surveillance and data mining which is often seen as an extreme invasion of privacy.[25]

An online example of what I refer to as anti-privacy code includes Gmail,[26] Google's free email service that scans all of your outgoing and your incoming emails for targetd advertising purposes.[27] At the time Gmail was launched, there was something of an outcry from privacy groups.[28] Not only is the privacy of the holder of the Gmail account impacted, but so to is the privacy of everyone the holder emails or receives emails. Thus, even if you consented to trading your privacy for the nice free email service, if you care about the privacy of your friends, you may hesitate to use Gmail. Another online example is what Sony BMG did when they covertly installed a rootkit into user computers via code installed when those users

[22] For a discussion of CCTV and its prevalence in the UK, see, for example, Robert D. Bickel, Susan Brinkley & Wendy White, *Seeing Past Privacy: Will the Development and Application of CCTV and Other Video Security Technology Compromise an Essential Constitutional Right in a Democracy, or Will the Courts Strike a Proper Balance?*, 33 STETSON L. REV. 299, 345 (2003).

[23] Gal Eschet, *FIPs and PETs for RFID: Protecting Privacy in the Web of Radio Frequency Identification*, 45 JURIMETRICS J. 301, 309 (2005).

[24] Anne Broache, *RFID Passports Arrive for Americans*, CNET NEWS, Aug. 14, 2006, http://news.cnet.com/RFID-passports-arrive-for-Americans/2100-1028_3-6105534.html.

[25] *See* EDWARDS, *supra* note 9, at ch. 16.

[26] Gmail: Email from Google, http://mail.google.com (last visited Dec. 27, 2009).

[27] About Gmail, http://mail.google.com/mail/help/about_privacy.html (last visited Dec. 27, 2009).

[28] *See* EDWARDS, *supra* note 9, at ch. 16; *see also* Hiawatha Bray, *Where are You Now?*, BOSTON GLOBE, Apr. 26, 2004, http://www.boston.com/business/technology/articles/2004/04/26/where_are_ you_now/.

bought Sony-manufactured CDs. Sony was pursuing the legitimate and crucial goal of trying to prevent its wares from being copied and shared in defiance of license conditions, but it used what has been described by some as an illicit method to reach this goal.[29] Normally, installing harmful rootkits and spyware without user notice and consent is regarded as a form of illegal computer hacking and/or actionable civil tort in most jurisdictions, regardless of the goal.[30]

Examples of pro-privacy code, outside the rather isolated village of specific PETS, include certain social networking sites (SNSs) or "blog" sites, like Live Journal,[31] that promote privacy as part of the service they supply. This idea is now mainstreamed (for example, it is in use on Facebook), but back in 2002, when Live Journal pioneered it, it was fairly uncommon. Live Journal's code allows a user to set up "friend lists" and make blog posts visible only by friends, or subsets of those friends. "Friends" here means people to whom you are prepared to make disclosure, people whom you trust, or people you are prepared to let into your private life. So I can use Live Journal to make posts that are public, meaning that everyone in the world can see them. I can also make what is called a "friends-only" post, whereby I can post information about my love life, medical condition, work place, and other personal details without making the post available to the entire world. These posts will only be readable by people that I put on my friends list. I can also create custom friends groups to control which friends receive certain information. You could have a custom friends group that you want to know about your love life, and it could be different than the group you wanted to know about your work or the group you wanted to know about your medical condition. Furthermore, there is a box you can check when you set up your Live Journal account to request that it not be searchable,[32] and then, no one can find you through Google or other search engines. This system is about trust building and community building, but it is also very sophisticated in allowing you the user to imbue the technology of the system with your own tailored privacy values. Over the last few years, despite the fact that social networking may seem inherently about seeking publicity, the ability to make friends-only posts has become a norm in the social networking world on multi-million user sites like Facebook and Twitter, as well as more minor hobbyist sites

[29] *See generally*, Deirdre K. Mulligan & Aaron K. Perzanowski, *The Magnificence of the Disaster: Reconstructing the Sony BMG Rootkit Incident*, 22 BERKELEY TECH. L.J. 1157 (2007).

[30] *See id.* at 1166–69.

[31] Live Journal, http://livejournal.com (last visited Dec. 27, 2009).

[32] A spider is a program that searches for information on the web by "crawling" websites. ZDNet, Spider: Definition, http://dictionary.zdnet.com/index.php?d=spider (last visited Dec. 27, 2009).

such as Live Journal. This phenomena has proved to be especially popular with older or more risk-averse users, particularly perhaps those with "alter egos," that is users with responsible public profiles who also have frivolous hobbies or alternate sexual tastes.

Another mainstream example of pro-privacy code relates yet again to spam. Spam, as I have previously stated, is a big problem. Pro-privacy code for guaranteeing your privacy and security against spam has been developed, and we now have very effective spam filtering.[33] My own spam filter system now has few false negatives, and only a few false positives. Again, we have seen a lot of coordinated private sector effort in this area. We had the anti-spam alliance with AOL, Microsoft, Yahoo, and Earthlink trying to work together to create an entrusted email system.[34] The IETF is working in this area as well.[35] Eventually, we should see some kind of trusted email, identified-sender system that effectively controls spam (since any email that disguises its origins can presumptively be discarded as likely to be spam).

We go back to that very simplistic idea discussed at the beginning of

[33] *See* FEDERAL TRADE COMMISSION, EMAIL ADDRESS HARVESTING AND THE EFFECTIVENESS OF ANTI-SPAM FILTERS 5–6 (2005), *available at* http://www.ftc.gov/opa/2005/11/spamharvest.pdf.

[34] *See* David Dickinson, Note, *An Architecture for Spam Regulation*, 57 FED. COMM. L.J. 129, 145 (2005).

[35] Two primary proposals were proffered to IETF for authenticating senders and limiting domain name spoofing and repudiation of spam: Sender ID and DomainKeys. SenderID was a proposal offered by Microsoft to authenticate the identities of senders of email and, thereby ostensibly, mitigate the spam problem. *See* Microsoft Corporation, SenderID Home Page, http://www.microsoft.com/ mscorp/safety/ technologies/senderid/default.mspx (last visited Dec. 27, 2009). Based on the sender's server IP address, SenderID was intended to eliminate domain name spoofing by preventing repudiation of email and holding ISPs accountable for spam sent through their services by confirming that each email message originated from the Internet domain it claimed. Consequently, recipients could seamlessly reject messages that claimed to be from an IP address that had not been declared by the alleged sender. *See* Brian Livingston, *Sender ID Declines, DomainKeys Shines*, DATAMATION, Sept. 28, 2004, http://itmanagement.earthweb.com/columns/executive_tech/article.php/3413611. Several critiques of SenderID were raised, including the fundamental incompatibility of its license terms with those of open-source products and a question regarding Microsoft patents of key SenderID technology, leading AOL, for example, to lose faith in the project. *Id.*; *see also Sender ID Loses Supporters*, ZDNET, Sept. 03, 2004, http://news.zdnet.co.uk/internet/security/0,39020375,39165420,00.htm. Currently the proposal appears to be fatally stalled in the IETF. DomainKeys, the proposal of Yahoo! and Sendmail, was termed a "cryptographic authentication solution" to the problem of phishing since it uses public-key cryptography to let users verify that a message actually comes from the domain that is listed in the sending address. *See* Yahoo! Media Relations, Sendmail and Yahoo! Collaborate to Develop and Deploy DomainKeys, http://docs.yahoo.com/docs/pr/release1143.html (last visited Dec. 27, 2009). Each ISP or email provider that implements the system has a private key that it uses to sign all outgoing messages and publishes its public key in the Domain Name System records. In this manner, DomainKeys would also certify that the contents of the message have not been altered in transit. Because all outgoing email servers would "sign" messages using a digital certificate, recipients could reject messages that did not comport with a listing in the World Wide Web registry. Although DomainKeys received more positive responses in the IETF and public debate than SenderID, the technology itself is now in question. *See, e.g.*, Dennis Fisher, *Scammers Exploit DomainKeys Anti-phishing Weapon*, EWEEK, Nov. 29, 2004, http://www.eweek.com/article2/0,1759,1732576,00.asp.

this piece that code is written to reflect the values of its owners and its writers. The owners and the writers of Microsoft, Yahoo!, AOL, for example, do not in all likelihood care much about the privacy of their users. However, Internet service providers (ISP) and host services, especially providers of webmail, such as Microsoft Hotmail, care a great deal about the congestion of the Internet because they spend a lot of time and money throwing away 90% of the email that runs through their systems.[36] Hotmail, for example, cares a great deal about spam because it gives their service a bad name and scares a very large number of users away. So these groups are not worried about privacy out of the goodness of their hearts, but the privacy of users is a spin-off of their concern for the financial stability of their enterprises.

The key question is why would owners and writers justify implementing anti-privacy code over pro-privacy code? For example, with regard to the Sony DRM rootkit,[37] I have asked myself, "How could Sony be so short-sighted?" And the straight answer appears to be that Sony was thinking about a primary economic goal of stopping IP infringement and not about their end user's privacy, security or trust in Sony. Their file-sharing concerns overwhelmed any concerns they had for the privacy or security of their users or the public.

CCTV[38] is an interesting example of a situation where the *public*, not just the code owners or writers, appear to favor anti-privacy code. There were many surveys in the UK during the 2000s that asked people whether they liked CCTV, and a high percentage of respondents usually said yes, because the general perception in the UK is that CCTV is what protects people from crime on the streets.[39] This is true despite empirical evidence, which actually shows that CCTV mostly moves the crime to a different area.[40] Thus, the public in this case is in favor of the anti-privacy code.[41]

[36] Debin Liu, Recent Development, *The Economics of Proof-of-Work*, 3 I/S: J/L & POL'Y INFORM. SOC'Y 337, 338 (2007).

[37] For a discussion of the Sony rootkit, see *generally* Bruce Schneier, *Real Story of the Rogue Rootkit*, WIRED, Nov. 17, 2005 http://www.wired.com/news/privacy/0,1848,69601,00.html. US-CERT, part of the US Department of Homeland Security advised consumers not to install software from an audio CD. Two sets of Sony DRM were implicated: XCP and SunnComm's Media Max version 5. *See, e.g.*, Brian Krebs, *Study of Sony Anti-Piracy Software Triggers Uproar*, WASH. POST, Nov. 2, 2005, *available at* http://www.washingtonpost.com/wp-dyn/content/article/2005/11/02/AR2005110202362.html.

[38] *See* sources cited *supra* note 22.

[39] Duncan Carling, Note, *Less Privacy Please, We're British: Investigating Crime with DNA in the U.K. and the U.S.*, 31 HASTINGS INT'L & COMP. L. REV. 487, 506 (2008).

[40] *See* Lisa Minuk, *Why Privacy Still Matters: The Case Against Prophylactic Video Surveillance in For-Profit Long-Term Care Homes*, 32 QUEEN'S L.J. 224, 265–66 (2007).

[41] However, this has changed slightly even in the UK as we have moved into the second decade of the 21st century and the public has been exposed to the reality of constant data breaches by the public

So far, RFID has seen little debate within the general public. Whether they will favour it as a security enhancing technology, such as allegedly stopping terrorists from entering the country by use of biometric passports, or whether there will be a privacy-conscious backlash as we saw with respect to the UK ID card after extensive campaigning by groups like No To ID,[42] remains to be seen.

B. Why Privacy-Friendly Code Matters

So why do people write privacy-friendly code, as opposed to privacy-invasive code? We have already briefly considered the above examples of spam filters, which act to save ISPs and webmail providers money, as well as to help build the brand and protect user privacy. Turning to the example of Live Journal, we next ask why did they choose to write into their code the capacity for users to protect their privacy via customizable friends groups, when the normal belief is that any SNS not charging for services will choose to collect as much data as possible from its users, and thus encourage disclosure and discourage privacy-enhancing code as much as possible?

Live Journal was, in conception, very much a grassroots, counter-culture kind of project. It is open-source code, meaning that anyone can use its source code to build their own blogging site (and this has happened, as with, GreatestJournal and DeadJournal). Importantly, its code was written by people who planned to use it themselves and to let others whom they regarded as friends or at least allies, to build upon it.[43] The early adopters were both writers and users, and that is, I think, one reason why it was built to be privacy enhancing. The people who wrote it valued their own privacy, security, as well as their sense of community, and wrote those values into the code. People do not necessarily sign-on to Live Journal because it overtly advertizes having strong privacy values, but they are attracted to the strong online community spirit, which is itself engendered by the fact that users and groups, especially those from alternative communities (such as Goths, gamers, alternate sexuality groups, BDSM groups, etc.) have tools available to protect themselves from the external gaze while still being able to network internally in quite complex ways. More customers thus amass,

sector.

[42] *See* Elitsa Vucheva, *EU to Launch Biometric Passports by Summer*, EU OBSERVER, Jan. 14, 2009, http://euobserver.com/9/27407.

[43] Note that Live Journal was however purchased by, Six Apart since the time this talk was given. For a discussion of open-source software, see generally Max Henrion, *Open-Source Policy Modeling*, 3 I/S: J.L. & POL'Y INFORM. SOC'Y 355, 357–58 (2007).

which means more goodwill and a higher price when the enterprise is sold. Thus, you can make privacy into a brand and a valuable feature (as was seen when Live Journal was recently purchased by Six Apart, a company previously specializing in social networking in the Russian market). Even now Live Journal is owned by an enterprise distinct from its users, and its clientele continue to police its privacy-enhancing features by making a fuss whenever there are signs that Live Journal is departing from the established community values, as was demonstrated when Live Journal decided to introduce a limited degree of advertising as the "price" of a free account. Live Journal constitutes an example of a company where privacy code was built into the system from the beginning as "privacy by design" and where such a function became part of its business model.

III. FINAL THOUGHTS

I have argued above that code neither inherently invades privacy nor preserves privacy; instead, code reflects the values of its writers and owners, and sometimes, where their views are crucial to economic success, its users. Can we use this insight to produce better privacy for users? Consumers will, experience shows, rarely lobby for privacy *per se*. In the UK, surveys have shown that in the e-services market, consumers see privacy as quite low on the list of buying features after things like price, accessibility, availability and convenience. But, if you can tie it to something consumers care about, or if you can make it something they care about, this may change.

The job of those who campaign for privacy online, therefore, seems to be to make privacy-enhancing be seen as a core value or as a feature, and not as a bug or an optional, after the fact, bolt-on extra. For the coders themselves, privacy, like security, is still most seen as something that gets in the way of business and is an overhead rather than something that makes money. Privacy gets in the way of money making activities like collecting data about users, which can be sold and used in targeting ads. Similarly, privacy regulation, such as the EU DP regime, is seen by most commercial concerns as yet more red tape and costly bureaucracy, rather than as something that helps companies produce a more appealing product. It is hardly surprising that DP is so often observed in the letter rather than in the substance and that despite seemingly tough privacy regulation, there have been so many data breach scandals in the UK lately.[44]

[44] *See* EDWARDS, *supra* note 9, at ch. 14.

Cory Doctorow has, however, suggested a different approach.[45] In the digital marketplace, it is often hard for one product to stand out. One MP3 download, for example, is much like another. He suggests that privacy might be sold as a key benefit of a product, something of interest to the public and enabling price discrimination. In a similar fashion, after public pressure, sites like iTunes have begun offering DRM-free MP3s at a premium cost. This is very much how I have suggested above that Live Journal gained a market niche. If you are a phone company, suggests Doctorow, you can gain customers by choosing to not keep customer activity logs indefinitely. If you are a search company, you can delete your cookies[46] and tell your customers about this practice. Once your customers get wind of the fact that the privacy they hoped for is for sale, and not unreasonably expensive, these entities may find a profitable niche.

This is the hopeful idea of "privacy as brand," though so far a rather optimistic one. Sadly, evidence to date seems to show that privacy has not yet become an obvious selling point in the mainstream. Indeed, in one study, researchers found a correlation between social network services which made a point of their privacy features in their publicity, and these sites performing poorly in the market.[47] However, much of this may be a result of the public lacking either the data or the skills necessary to discriminate on grounds of privacy. As data privacy scandals have rocked the UK and the U.S. in the last few years, it is quite possible the public will begin to be incentivized to acquire these skills and demand such data, thus creating a market for privacy.

My final point is that in relation to privacy and code, technology defaults are crucial. Facebook is a good example of this. Facebook often asserts that it provides a huge variety of privacy enhancing tools to its customers and this is in fact true. But in general, the *defaults* of the site are set so that the user is encouraged to disclose the maximum data, and keep the least amount of control. For example, it is not inconceivable that profiles could be set to be readable by "friends only" until the user chooses to change the setting to "public." Likewise a new user could be forced to work through a brief "set-up" routine before starting to use the site, akin to an "install" routine, in which they could be encouraged to decide how to set

[45] *See* Cory Doctrow, *Ethics Are the New Craft*, 1 SCRIPTED 514 (2004), *available at* http://www.law.ed.ac.uk/ahrc/SCRIPT-ed/issue4/cory_ed.pdf.

[46] ZDNet, Cookie: Definition, http://dictionary.zdnet.com/definition/Cookie.html (last visited Dec. 27 2009).

[47] *See* Joseph Bonneau, *How Privacy Fails: The Facebook Applications Debacle*, LIGHTBLUETOUCHPAPER, June 9, 2009, http://www.lightbluetouchpaper.org/2009/06/09/how-privacy-fails-the-facebook-applications-debacle/.

their privacy controls in an understandable way to their own satisfaction. Privacy controls on Facebook are currently hard to find, are not changeable in a consistent way, and so are hard to master.These privacy controls become more difficult in each new iteration of the Facebook desktop.

The reason for this is simple: Facebook makes its money not from user subscriptions but from selling user data to marketers. More privacy means less available data, and subsequently less money available for Facebook to make by selling that data. Accordingly, by market failure, it is possible that defaults may have to become the subject of regulation,[48] although this takes us into the difficult terrain of "technology mandates." Nonetheless, in the EU at least, there is considerable regulatory concern about this matter, especially in relation to young and vulnerable people who are the predominant users of social networking sites.

Controversially, this writer, who was raised working in the European regulatory tradition, does believe that we need law to intervene here. We need scrutiny of code in socially sensitive applications, especially in the form of technology defaults, if we are to ensure that democratic and privacy values are built integrally into the next generation of code, despite the fact that market pressures will often—if not always—push towards unfettered disclosure. How this scrutiny and regulation should be achieved is a major challenge for the future. One interesting development is that in Europe, the long established Data Protection Directive is finally coming up for review, and Viviane Reding, the Commissioner in charge, has given an early indication[49] that she is in favour of regulation to support "privacy by design." If this survives market onslaught to find its way into the new Directive, it will be a radical victory for those who wish to see more deliberate creation of privacy-enhancing code in mainstream applications. This interesting space should be watched by the EU and U.S. alike, especially considering that an initial draft of reforms is expected by end 2010.

[48] *See* Lilian Edwards & Ian Brown, *Data Control and Social Networking: Irreconcilable Ideas?, in* HARBORING DATA: INFORMATION SECURITY, LAW AND THE CORPORATION (Matwyshyn, A. ed., 2009) *available at* http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1148732.

[49] Press Release, Viviane Reding Vice-President of the European Commission Responsible for Justice, Fundamental Rights and Citizenship Next Steps for Justice, Fundamental Rights and Citizenship in the EU European Policy Centre Briefing, Brussels (Mar. 18, 2010), *available at* http://europa.eu/rapid/pressReleasesAction.do?reference=SPEECH/10/108.