International Review of Law Computers & Technology, Volume 17, No. 1, Pages 85–97, 2003

Carfax Publishing
Taylor & Francis Group

Regulating Architecture and Architectures of Regulation: Contributions from Information Systems

IAN HOSEIN, PRODROMOS TSIAVOS and EDGAR A WHITLEY

ABSTRACT *Lawrence Lessig's argument that 'code is law' runs the risk of reifying the notion of code. This paper balances Lessig's view about the plasticity of technology with experiences from information systems. By considering two diverse cases: peer-to-peer file sharing software and the cryptographic interface in Microsoft products, the paper presents a revised understanding of the relationship between technology and regulation.*

Introduction

In a world where technology and politics interact, there is increasing discussion of the role of law and government action. This is particularly true in the case of cyberspace, where the Internet is changing the ways in which regulation is applied. In the extreme case, technology is claimed to determine freedom and rights. This position has been addressed by Laurence Lessig, in his book 'Code and other laws of cyberspace'.[1]

In this book, Lessig discusses how regulation could occur, and does occur, within a new regulatory habitat[2] where a technology's architecture (or 'code') acts alongside the traditional regulators of the market, laws, and norms. According to Lessig, code can be an effective regulatory mechanism, but also represents the interests of its developers and is therefore socially shaped.[3]

Correspondence: *Edgar A Whitley, Department of Information Systems, London School of Economics and Political Science, Houghton Street, London WC2A 2AE, UK. Web: http://is.lse.ac.uk; e-mail: {I.Hosein; P.Tsiavos; E.A.Whitley}@lse.ac.uk.*

Ian Hosein is a Visiting Fellow and PhD candidate in the Department of Information Systems at the London School of Economics and Political Science. Components of this research were made possible by a grant from Microsoft Research, UK. For more information please see http://is.lse.ac.uk/staff/hosein. Prodromos Tsiavos is a PhD Candidate in the Department of Information Systems at London School of Economics and Political Science and is currently funded by the Greek Scholarship Foundation for conducting research on Information Technology Law. Edgar A Whitley is a Senior Lecturer in Information Systems at the London School of Economics and Political Science.

ISSN 1360-0869 print/ISSN 1364-6885 online/03/010085-13 © 2003 Taylor & Francis Ltd
DOI: 10.1080/1360086032000063147

Although Lessig highlights the fact that code is designed and hence he holds a position that runs counter to notions of strong technological determinism, he still seems to reify the notion of code. Information Systems studies the development, implementation and use of computer-based systems and offers an alternative reading of code from that provided by Lessig. By exploring code in more detail, it is possible to add the same level of sophistication to our understanding of how code shapes regulation as Lessig has done for the other elements of his framework for regulation.

Examples of areas that Information Systems has studied that apply to Lessig's problem area include the differences between bespoke and packaged software[4] and the unintended consequences of large-scale software implementations.[5] Other research has shown the implications of low-level code design decisions on organizations and industries. For example, Scott[6] describes the transformation of risk in financial markets arising from the introduction of a new system to aid in loan decisions.

In this paper, we seek to develop questions of regulation through architecture by presenting two cases where the design of software systems affects the ways in which regulation can take place. First, we consider the design of peer-to-peer (P2P) networks for file sharing systems; second, we review Microsoft's Cryptographic Application Program Interface (CAPI). In so doing we challenge the reified notion of architecture as a straightforward aspect of regulatory intervention.

In order to structure our critique of code, we introduce three main ways in which to consider technology and regulation. These are the ways in which technology can object, the modalities of regulation and the path of technology regulation.

Technology as a Regulatory Actor: Objecting

In both of the cases of regulation involving technology that will be presented in this paper, the construction and the constitution of the technology are important components in the regime. Within the literature on regulation however, the role of technology is often poorly attended to. While it may be acknowledged as a disruptive force, as Peltzman notes within telecommunications rate structures[7] and interest rate regulation,[8] it is not investigated in detail. The need to consider technology in regulation is not new, however, nor is it particular to the Internet and digital communications technologies; as Levin[9] noted regarding the management of the radio spectrum in the 1960s: 'New communications has required the re-examination of many policies and assumptions in recent years'.

The ability to examine technology policies and regulatory change is limited, as studies usually concentrate on human action. Hood[10] notes the many ways of seeing regulatory change: the power of interest groups, power of ideas, social transformations, or a policy's self-destruction;[11] these explanations are all assuming that regulatory change is enacted only by humans. As Peltzman notes, pressures for deregulation include:

> … changes in the 'politics' and changes in the 'economics' of the regulated industries. Political change includes such things as shifts in the relative political power of contending groups and changes in the underlying organization and information technologies.[12]

Once again technology appears to be secondary to powers of humans and group interests. Meanwhile, Porter and van der Claas[13] acknowledge that a failure in some environmental regulations is the static view of technology, and they believe that ideal regulation would set the conditions for the creation of technology to aid environmental interests. All of these concepts fail to acknowledge that technology is a strong regulatory factor in itself, and they fail to look in detail at how technology may disrupt or promote regulation.

The disruptive capacity of technology has been noted previously within the social studies of technology literature. Technologies may disrupt human action through objecting.[14] Latour's view is that technologies can force us to renegotiate our paths and goals,[15] and so restructure our regulations. They can also resist our attempts to regulate and deregulate, possibly more so than humans and institutions.

Technology and Human Action: Habitats and Modalities

Technology as a regulatory factor is not a new idea; but requires interrogation. Often when technology is brought to the foreground for analysis of a societal issue, it is considered a deterministic force, ignoring the power of human action. In academic studies it is poorly investigated and interrogated, often left in the background of analysis. At best technology is considered as a quiet part of the policy habitat,[16] in what is considered a society-centred approach, where policy changes arise due 'to the background changes in technology and social structure'.[17] As the social structure or the technology changes, it is said, there is a resulting loss of policy habitat, resulting in regulatory change. Our goal is to find a way of looking at the policy habitat as a socio-technical phenomenon, where the technology can be brought to the forefront, alongside the social actors.

Such technological and policy habitat explanations of policy shifts are considered problematic for a number of reasons. First, we must avoid views where technology is seen as deterministic; regulatory change is not automatic from technological change.[18] Hood continues that even when similar policies are adopted by different societies, 'it does not necessarily mean that they are responses to the same "functional" problems'; and 'social change is not necessarily an independent factor from which everything else stems—it may itself be the product of other policies, designed to "shape" preferences'.[19] Hood concludes that previous attempts to explain the loss of habitat give too much credit to technology, 'leaving too little room for the autonomous dynamics of politics'. Again, society and technology theorists are well aware of these concerns; as this is consistent with the anti-essentialist theory that argues we must never assume that technology is an object in itself with its own capacities; but rather these capacities are always interpretations.[20] Technology and regulation therefore interact within the realm of politics; different societies will interpret the problems and the technologies differently. Change may (or may not) come about due to technological or political issues.

Therefore a richer view is required; technology is not deterministic, as it requires human interpretations and action. Yet, human action is not deterministic either, as technology objects and also regulates human action through the loss of a policy habitat. Lessig tried to develop a view where technology, law, norms and the market were all modalities of regulation[21] that constrain individuals.

> There were times these other constraints were treated as fixed—when the constraints of norms were said to be immovable by governmental action, or the market was thought to be essentially unregulable, or the cost of changing real-space code was so high as to make the thought of using it for regulation absurd. But we see now that these constraints are plastic. That they are, as law is, changeable, and subject to regulation.[22]

Lessig establishes that all of these *modalities of regulation* can all be shaped; but in his zeal to dismiss determinism and to say that technology can be shaped (it is only plastic like the other modalities), he misses two points. First, he fails to acknowledge that the other modalities may interpret the technology in a number of different ways. Which laws apply, how many definitions of a given application exist in law? For example, cryptography is in law as a confidentiality tool and a tool for digital signatures. Each modality does not necessarily understand technology in the same specific way. Second, in his claim that the constraints are merely plastic, he does not acknowledge how technology may object, how it refuses to be shaped by the interests of the other modalities. This latter point will be discussed in greater detail in the next section.

Walking the Path of Regulatory Discontinuity

When Lessig analyses *the four modalities of regulations*[23] he makes a distinction between architecture and the other three modalities, namely law, the market and norms: 'The constraints of architecture are self-executing in a way that the constraints of law, norms, and the markets are not'.[24]

Analysing in more detail his idea of *self-execution* of architecture as a regulatory modality, Lessig provides two further criteria, namely *agency* and *temporality*, based on two perspectives: 'that of someone observing when a constraint is imposed (the objective perspective), and that of the person who experiences the constraint (the subjective perspective)'.[25]

From the objective perspective, Law and norms always have an *ex post* sanctioning effect: if you violate the rule, the sanction comes *after* your action. On the contrary, Market and Architecture have an *ex ante* preventive effect: you cannot perform the action at all.

From the subjective perspective, the adherence to the rules contained in each one of the four modalities the acting subject is internally prevented from doing a particular action. The more the internalization process is advanced the less probable it is that the person is going to perform the prohibited action and thus the more effective the regulatory modality is. For Lessig, law and norms are dependent on the internalization of the rule, whereas this is not the case for architecture. Quoting Sorkin, Lessig describes the essence of code's regulatory capacity: 'Whatever the sources of the content of these codes, ... their consequences are built'.[26]

Code has indeed a great regulatory capacity; at the same instance it is a highly idiosyncratic form of regulation and we cannot really assess its consequences unless we attempt to analyse it.

First and foremost, technology as a regulator is not deterministic. Although Lessig overemphasizes the self-execution of the code as a regulatory modality he also acknowledges that there is no certainty that a technology will produce a particular behaviour; rather it will affect it.[27] This realization has a profound effect, as it highlights the subjective or soft element of architectural regulation. Lessig claims that 'architecture can constrain without any subjectivity',[28] but in the same book he urges us to realize the 'plastic' nature of technological constraints:[29]

> There are choices we could make, but we pretend that there is nothing we can do. We *choose* to pretend; we shut our eyes. We build this nature, then are constrained by the nature we have built.[30]

The regulatory nature of architecture starts long before it is in place. The regulatory nature of architecture lies beyond its 'artefactual' manifestation and is deeply rooted in human subjectivity as can be seen in the quotations of different users of P2P technologies that we present in the case study.

The construction of the regulation does not solely have to do with the code; it is inherently linked to the use and implementation of the code as well. As the Actor Network Theory theorists have often emphasized, 'reality is a process'[31] or to use Latour's aphorism 'For technology, every day is a working day'[32]. We need to move beyond the mere observation that technological constraints are 'plastic' and constructed and try to explore the mechanics of this construction.

To advance the argument on the non-deterministic nature of technology we need to understand that technology has always a series of unintended consequences that sometimes have a greater impact than the original intended plans of its creators.[33]

Case Study 1: Napster

In May 1999, Shawn Fanning, an undergraduate student at Northeastern University, created an application called Napster. The idea behind Fanning's software was to enable end-users to share the MP3 files stored in their computers, using a centralized indexing service to locate the files. Two years after its launch, Napster had experienced an exponential growth to reach an audience of over 50 million users. Napster's popularity resulted in a lengthy legal battle between the music industry and Fanning's newly founded company on issues of copyright infringement.[34]

Perhaps unsurprisingly, the Napster case contributed towards the software's notoriety and led to the establishment of 'Napster' as a generic term for P2P networks. Fanning's file-sharing service, however, was neither the first P2P technology used for file sharing, nor was it a pure P2P application. Indeed, what is meant by P2P is unclear:

> Taken literally, servers talking to one another are peer-to-peer. The game Doom is peer-to-peer. There are even people applying the label to e-mail and telephones. Meanwhile, Napster, which jump-started the conversation, is not peer-to-peer in the strictest sense, because it uses a centralized server to store pointers and resolve addresses.[35]

If Napster was not a 'pure' P2P application and it is not clear what P2P really is, then why insist on using the term for explicating the regulatory properties of technology?

Regardless of the technical nature of Napster, it has been inextricably linked to the term 'peer-to-peer' and this has led to the labelling of a whole family of technologies as anti-regulatory, anti-authority devices. The social and conceptual construction of P2P technologies as anti-control mechanisms was a collective process facilitated in part by the media industry and the high profile legal action against P2P services that made them an icon for teenage users and thus triggered their proliferation, sophistication and anti-authority status.

Most of the P2P services have been entangled in legal disputes with the media industries. Napster was sued in 1999 and a shortly thereafter Scour faced a very similar fate. Both companies had to suspend services (from July 2001 and December 2000 respectively). In the midst of a series of legal developments and extensive media hype, the Recording Industry Association of America (RIAA) and Motion Picture Association of America (MPAA) have also gone after a number of other P2P services based on the most advanced P2P technology available at the time: that provided by FastTrack. Morpheus, KaZaA, Xolox and Grokster have respectively been the targets of the media industry both in the US and in

The Netherlands. Xolox was shut down in the process, but Morpheus and KaZaA have survived to become the two largest P2P networks. RIAA and MPAA have also expressed their intention to prosecute individual P2P users.

Commentary on Napster

The whole process of the interaction between the authorities and P2P technologies can be seen as an example of the 'cockroach phenomenon' (Figure 1): after the filing of lawsuits and the subsequent shutting down of Napster and Scour, a new generation of P2P services came to the fore. Many of these had already been around for some time; however, it was the enforced suspension of Napster that made users migrate to them. The so-called 'Napster Diaspora' soon became as large as the original 'Napster Community'. Although metrics for P2P use are not readily available, it is estimated that about 109 million users have downloaded Morpheus and over 195 million have the KaZaA desktop.

The RIAA and the MPAA responded with a second round of lawsuits against companies using FastTrack technology. The result was not entirely successful. Some of these companies were shut down but those that survived gathered most of the remaining users. In addition, other services that were not under legal prosecution appeared and in 2001 hosted over 75 million users. In a sense the legal battles functioned as a catalyst in the process of the physical selection of different technologies and disturbed the existing P2P 'ecosystem'; the closure of Napster caused users to seek alternatives; by publicising the whole process (and once you start a legal battle in the media sector it is impossible not to go public) Internet users were made aware of P2P technologies and the user base increased.

Most importantly, however, the phenomenon is both quantitative and qualitative. The new P2P systems were not simply more; *they were different from the earlier systems*. Only the more technically sophisticated and legally resilient systems could survive the RIAA/MPAA attacks. Moreover, all the P2P systems that came into existence during or after the Napster and FastTrack cases, irrespective of whether they had been a direct target of the

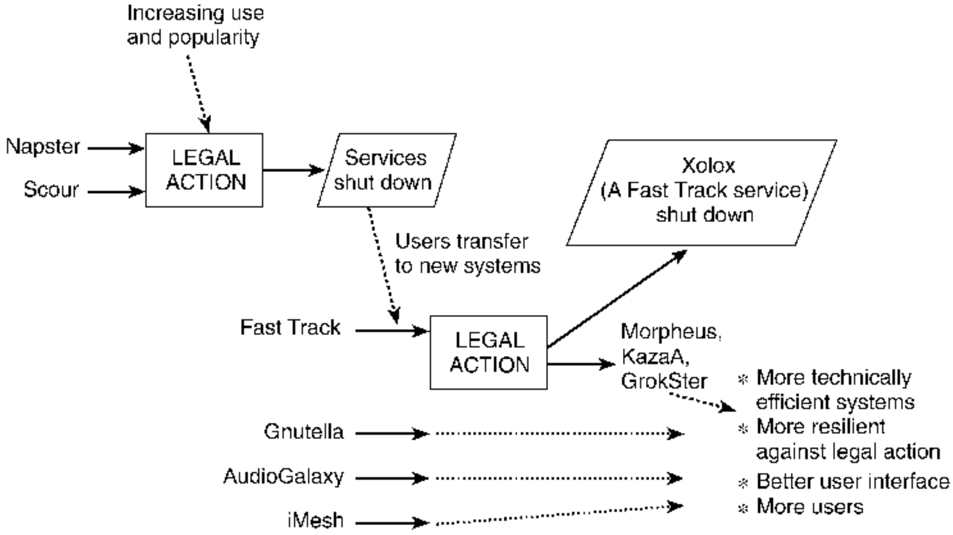
Increasing use and popularity
Napster
Scour
LEGAL ACTION
Services shut down
Users transfer to new systems
Xolox (A Fast Track service) shut down
Fast Track
LEGAL ACTION
Morpheus, KazaA, GrokSter
Gnutella
AudioGalaxy
iMesh
* More technically efficient systems
* More resilient against legal action
* Better user interface
* More users

Figure 1. The cockroach phenomenon.

media industry or not, encapsulated the characteristics of the systems that survived those legal actions: they had a bigger degree of decentralization of the indexing service and they were either open-source or located in P2P-friendly jurisdictions.

Another unanticipated consequence of the legal action against P2P systems was the emergence of the P2P *concept* itself. P2P existed as a technology for many years before the appearance of Napster. However, as we have already indicated it was the music-file sharing systems that popularized the technology and it was the hype surrounding the legal actions against it that gave P2P its current notional content and an anti-authority focus.

However, by emphasizing the anti-centralization and anti-hierarchical attributes of P2P technologies, it is possible to adopt a rather one-sided view of the whole issue. Whilst P2P technologies have characteristics that defy sources of external control, at the same time they are very much about 'awareness of the self and the others';[36] they are about sharing, interconnectivity, parallel processing, communication and collaboration. Each of these goals can only be achieved through the use of rules governing the operations and relations between the nodes of P2P networks. That is, by their very nature, P2P networks are self-organizing and self-regulating:

> Obviously much remains to be done before P2P establishes itself as a lasting force. System monitoring, remote peer control, usage metering, and accounting methods are just few of the areas that need further research.[37]

Thus the continued successful development of P2P networks will require the accommodation of more self-regulatory mechanisms. The technology that is the architect of freedom to share files becomes a technology that enforces self-regulation.

In a very elementary level these mechanisms are needed for the solution of seemingly pure technical problems. It is nevertheless impossible to distinguish the regulation of technical issues from the regulation of human behaviour, especially when the latter occurs within the framework of these systems.

For instance, almost all of the second-generation file-sharing systems allow the users to regulate the number of downloads and uploads that will take place in their part of the network. Also, in order to retain their operation many of the P2P file-sharing service providers install monitoring software.

The whole issue requires further research to identify and explicate the regulatory characteristics of P2P systems. At this point we just need to emphasize the fact that efforts externally to control technologies as disruptive as P2P systems are inherently flawed and can only lead to the cockroach phenomenon. At the same time, the position that P2P systems are anti-regulatory or anti-control is too simplistic and crude to be accepted. They encapsulate regulatory characteristics that control both technology and human behaviour. The point is not whether regulation exists or not but which is its locus and content.

Case Study 2: Microsoft, Export Controls and CAPI

Throughout the 1990s, the USA has revised its controls on the development and proliferation of *cryptography*, the means of establishing confidential communications and data to ensure integrity and authenticity.[38] The primary mechanism for controlling cryptography was export control regulation on products. This section describes the regulatory environment in the 1990s in which Microsoft developed its cryptographic solution, CAPI, up to 2000.

The substance of the regulations gradually changed from crude export controls on ‘munitions’ to a more sophisticated understanding of the nature of the technology. In particular, the relative *strength* of cryptography implemented in products has been controlled by requiring a reduced *key size*—the relatively unique numbers that lock and unlock the data—to a size that can be *brute-forced*—guessed within a reasonable amount of time. While recommended key sizes for symmetric key cryptography are at least 128-bits long, ie 2^{128}, the US Government regulated the size of exportable key-sizes to between 40 and 56 bits, in various cycles of deregulation. These shorter key-sizes are far easier to brute-force than the 128-bit recommended standard.

One consequence of the cycles of shifting regulations in the late 1990s was to create an unstable environment for code writing, especially for commercial packages. A company, such as Microsoft, which hoped to sell its products in many different markets needed to produce many different versions of the software to comply with the export controls that applied at any given time.

For many software developers, this required developing two pieces of software: one for domestic use where 128-bit + keys were enabled, and another for export where reduced key sizes were enforced. Distributors with international markets were in an inconvenient situation, and this was sometimes costly in terms of production and public relations.[39] An alternative was to sell one version with *weak* encryption for both domestic and international use.

Microsoft was susceptible to these concerns, having to cater for both the US export controls and consumer concerns regarding on-line security.[40] To deal with the regulatory environment, Microsoft embarked on finding a solution that could be designed and implemented in all its products, but that also satisfied export restrictions.

Dealing with such problems is common in software development and the standard solution is to separate out the functionality that changes, from functionality that remains relatively stable. By providing a standard interface to the variable aspects of the system, developers are isolated from the changing environment. This can be seen, for example, in the design of internet software, where the standard design of a web page is done without any consideration of the means of transmitting data over variable aspects of cables and networks.

Microsoft implemented its solution to the problems of dealing with changing regulations through the development of its Cryptographic Application Program Interface (CAPI) that acts as standard interface to cryptographic software. By using CAPI, the developers of an application simply had to call the appropriate routines without worrying about how they were implemented. The cryptographic modules, or Cryptographic Service Providers (CSPs) achieve the flexibility and could be adapted to address the changing export controls by including different key sizes, algorithms and operations to be processed.

The benefit of this design was that one set of code (CAPI) is implemented for worldwide distribution and then export restrictions would apply to the CSPs. Microsoft removed the liability for export by implementing cryptography into CAPI, as the functionality is enabled by CSPs, that are not necessarily Microsoft products.

Commentary on CAPI

However, not just any CSP would work with CAPI. The architecture of the system was refined to allow for a particularly interesting form of regulatory intervention. In particular, the US Government was concerned that outside software developers should not be able to write their own CSPs that would work with CAPI. It therefore required that CAPI first check that the CSP it was planning to operate with was an approved piece of software.

> … export control laws not only constrain cryptographic and related products but also any products which are specifically designed to interface to, or integrate with, cryptographic products. In effect, therefore, the very principle of openly available (CAPIs) is in direct conflict with the existing export control provisions in many countries. Thus, to integrate a CAPI into their operating systems without making them subject to export control Microsoft has had to establish some rigorous CAPI control procedures.[41]

CAPI was thus designed so that it would only work with CSPs that are digitally signed and verified by a public signature key embedded within CAPI. In order to be digitally signed in this way, however, the CSP had to be approved by the US government. An unsigned CSP or forged-signature CSP would not operate with CAPI because the CSP authorization-verification procedure would fail (see Figure 2).

As Gladman makes the case for Microsoft:

> It is important to recognize that this situation is not of Microsoft's making. In publishing and promoting a CAPI for use with their products Microsoft has gone as far as it can under US law to establish an improved basis for the provision of cryptographic information security when using their products. The procedures … are the provisions which the United States administration has imposed in order that Microsoft can offer their operating systems in world markets without being subject to US export controls.[42]

Every CSP developer would have to gain approval for export from the US government and only then would Microsoft digitally sign the CSP.

> Without such a signature requirement, there is no way for Microsoft to guarantee a CSP is staying within export guidelines. Because unrestricted access to CAPI would make Windows ineligible for export, the signature requirement limits CAPI access to vendors that agree to implement in conformity to US law.[43]

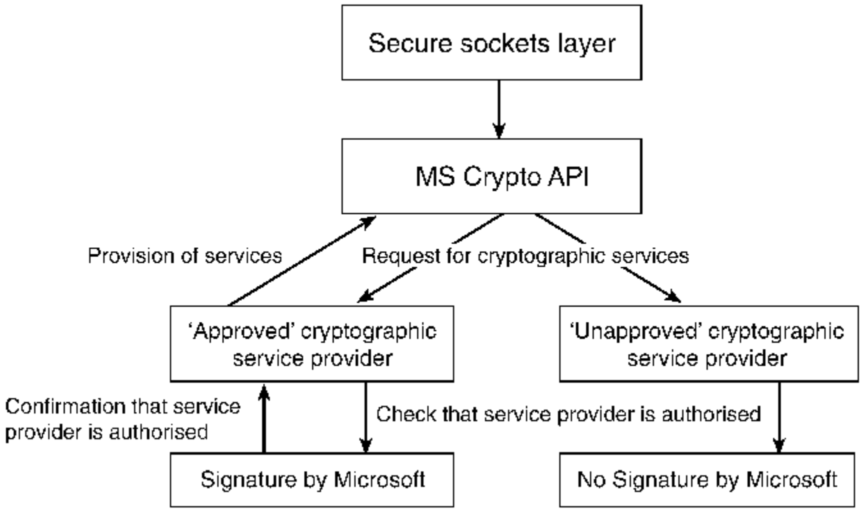
Secure sockets layer
MS Crypto API
Provision of services
Request for cryptographic services
'Approved' cryptographic service provider
'Unapproved' cryptographic service provider
Confirmation that service provider is authorised
Check that service provider is authorised
Signature by Microsoft
No Signature by Microsoft

Figure 2. CAPI with two CSP plug-ins, one being non-signed CSP.

The signing of the CSPs may be interpreted differently, however. According to Microsoft:

> The primary purpose of the digital signature is for the protection of the system and its users: by signing the CSP the integrity of the CSP can be guaranteed to the operating system. The operating system validates this signature periodically to ensure that the CSP has not been tampered with.[44]

The signing process, therefore, appears to act both as a gatekeeper for Microsoft and as a regulatory-enforcement mechanism for the US government.

The regulatory burden is thus placed on the CSP developers. First, they need to get a CSP software developers kit (CSP-SDK); if a developer is outside of the US, they must apply for a license, which is in turn also regulated by the US Government.[45] The foreign developer would still have to acquire US government approval before the CSP can be used with CAPI. Gladman summarizes this situation:[46]

CSPs Produced in North America for Domestic Use

> *The CSP Software Development Kit (SDK) is freely available without export control.* Microsoft will sign a CSP module without US (or other) government involvement.

CSPs Produced in North America for Export

> *The SDK is freely available without export control.*
> Microsoft will sign a CSP module given evidence of United States government export approval.

CSPs Produced Outside North America

> *The SDK is subject to US government export control.*
> The Microsoft signature on a CSP is deemed to be a 'defense service' provided by Microsoft to an overseas supplier and as such it is subject to the provisions of United States export control laws.

In effect, US export controls applied to all products that are designed for CAPI, whether they were developed in the USA for export, or developed in Europe. This extra-jurisdictional reach is a design implication of CAPI and invites the US Government into the functional loop of authorizing the operation of applications with Microsoft Operating Systems.

While there are no indications that CAPI was designed to lock out non-North American CSP developers, that they needed to get approval by US export regulators was an obvious hindrance. If foreign CSP developers wished to make strong cryptography available to non-North Americans, the recommendation articulated from Microsoft, as documented and analysed by Gladman, was:

> 'For suppliers who want to maintain the same product across all markets, North American and everywhere else, the most attractive strategy remains to develop CSPs outside the US or Canada and outside (CAPI).' This is again a clear recognition on Microsoft's part that it will NOT be possible to use their CAPI to support the general availability of strong cryptography outside the United States and Canada.[47]

Gladman concludes that Microsoft expected the US government to use CAPI to limit the development and use of cryptographic capabilities outside the USA.

> In terms of practical effect the mechanisms for the control of CSP signatures will be used by the United States administration to extend the scope of US export controls to cover CSP modules produced for domestic use in other countries even when there is no legal basis for such domestic control either in the United States or in the country concerned.[48]

Under that export regime, while North Americans had open access to strong cryptography with Microsoft products, 'the rest of the world will have nothing of any real value except in specialized application approved by the United States administration'.[49] Gladman later noted[50] that a UK office was later set up for the signing of CSPs.

Conclusions

With technologies increasingly developing regulatory features, the boundaries between regulation and enforcement are gradually blurring. Are we regulating technology or are we regulated by technologies? Taking up Lessig's call for research, 'In cyberspace we must understand how code regulates—how the software and hardware that make cyberspace what it is regulate cyberspace as it is' we have presented two cases that will help us overcome the reified notion of code.

These are now analysed using the themes presented earlier for refining our understanding of code and other forms of technology.

Technology as Objectors

P2P systems may be seen as technologies that object to or resist regulation, even as companies fold to legal pressures. CAPI may resist deregulation; that is, even after the US export controls changed under pressure from lobbyists and privacy advocates and thus Microsoft would no longer be required to review the CSP export permits, CAPI would continue to require the digital signature of Microsoft to function, still disallowing all now legal non-signed CSPs. As a result, technology can act to bring about change through objection and resist deregulatory efforts even as human institutions bend.

Habitat and Modalities

Lessig wished to counter the technologically deterministic view that the Internet cannot be regulated. Instead, he leaves us believing that technology is merely plastic and by extension can be shaped by the other modalities of regulation. While his conception of the policy habitat is rich, he does not allow for technology's ability to object to new policies, nor its ability to sustain intentions that are not embedded by the market, laws, or norms. Nor does he allow for how the other modalities of regulation will interpret the technology in different ways (eg the market views cryptography differently than the law). CAPI's requirement for a Microsoft signature across jurisdictions regulates regardless of the market, laws and norms.

The Path of Regulation

Moreover, there is no such a thing as a unique monolithic technology. Instead we are always faced with families of technologies that develop their internal dynamics and evolve in unanticipated ways. For instance, AudioGalaxy may impose a kind of regulation by installing by default a spy-ware application on the user's hard drive. However, technologies like Ad-Aware allow the user to override the spy-ware applications. If the norm of AudioGalaxy and Gator is 'you have to pay with your personal data for the services we provide', the norm of Ad-Aware is 'information is free, personal data has to be protected'.

The 'cockroach phenomenon', which we described in the case study, illustrates that there is no certain outcome in the interaction between different modalities of regulation, neither is one modality totally separate, independent or more effective than the others.

Revisiting Lessig's conceptualizations about technology as a regulatory factor, we see the mirage of control to be collapsing. Indeed technology is a modality of regulation, but it is one that cannot really be separated from the other modalities nor does it make any sense without them; it is not clear who is the creator of this regulation as it contains inscriptions from various (and sometimes undefined) actors; the content of the regulation is not always clear and it keeps changing through time; it does not have a deterministic effect and sometimes it is even autonomous.

Is technology indeed a form of regulation? We hold that it has *regulatory characteristics*, but the question of whether it constitutes a modality of regulation remains open. Tracing the regulatory characteristics of technology is exploring a path of discontinuities. Even if technology *is* regulation, it is of a very special kind. Before we have traced and analysed some of its characteristics we cannot say anything truly meaningful about it.

Following from Lessig and Hood, we believe that in the foreground of our techno-regulatory analysis we need to have both societal and technological factors, interacting, interpreting and objecting. If we see regulation as a socio-technical issue, with human actors inscribing interests into regulation and technology, interpreting the technology, but the technology being able to object, then we may better understand how a policy habitat is developed, sustained and lost.

Notes and References

[1] L Lessig *Code: and Other Laws of Cyberspace* Basic Books, New York, 1999.

[2] C Hood *Explaining Economic Policy Reversals* Open University Press, Buckingham, 1994.

[3] D MacKenzie and J Wajcman (eds) *The Social Shaping of Technology* Open University Press, Buckingham, 1999.

[4] E Carmel and S Sawyer 'Packaged software development teams: what makes them different?' *Information Technology & People* Vol 11, No 1, pp 7-19, 1998.

[5] CU Ciborra & Associates *From Control to Drift: The Dynamics of Corporate Information Infrastructures* Oxford University Press, Oxford, 2000.

[6] S V Scott 'IT-enabled Credit Risk Modernisation: A Revolution Under the Cloak of Normality' *Accounting, Management and Information Technologies* Vol 10, No 3, pp 221-255, 2000.

[7] S Peltzman 'The Economic Theory of Regulation after a Decade of Deregulation' in R Baldwin, C Scott and C Hood (eds) *A Reader on Regulation* Oxford University Press, Oxford, 1998, p 117.

[8] *Ibid*, p 121.

[9] H J Levin 'New technology and the old regulation in radio spectrum management' *The American Economic Review* Vol 56, No 1/2, pp 339–349, 1966.

[10] Hood, *op cit*, note 2.

[11] *Ibid*, p 4.

[12] Peltzman, *op cit*, note 7, p 108.

[13] M E Porter and L van der Claas 'Toward a new conception of the environment-competitiveness relationship' *The Journal of Economic Perpsectives* Vol 9, No 4, pp 97–118, 1995.

[14] B Latour 'When things strike back: A possible contribution of science studies to the social sciences' *British Journal of Sociology* Vol 51, No 1, pp 107–124, 2000; C Sørensen, E A Whitley,

S Madon, D Klyachko, G Hosein and J Johnstone 'Cultivating recalcitrance in information systems research' in N Russo, B Fitzgerald and J I D Gross (eds) *Realigning Research and Practice in IS Development: The Social and Organisational Perspective* Kluwer, Boise, ID, pp 297–316, 2001.

15 B Latour *Pandora's Hope: Essays on the Reality of Science Studies* Harvard University Press, Cambridge, MA, 1999.

16 Hood, *op cit*, note 2.

17 *Ibid*, p 10.

18 *Ibid*, p 12.

19 *Ibid*, p 12.

20 K Grint and S Woolgar *The Machine at Work: Technology, Work, and Organization* Polity Press, Cambridge, MA, 1997.

21 Lessig, *op cit*, note 1, p 88.

22 *Ibid*, p 91.

23 *Ibid*, p 235.

24 *Ibid*, p 236.

25 *Ibid*, p 237.

26 *Ibid*, p 239.

27 *Ibid*, p 239.

28 *Ibid*, p 238.

29 *Ibid*, p 91.

30 *Ibid*, p 234.

31 M Callon 'Some elements of a sociology of translation: domestication of the scallops and the fishermen of St Brieuc Bay' in J Law (ed) *Power, Action and Belief: A New Sociology of Knowledge?* Routledge & Kegan Paul, London, 1986, pp 196-233.

32 B Latour *Aramis, or the Love of Technology* Harvard University Press, Cambridge, MA, 1996.

33 Ciborra & Associates, *op cit*, note 5; Latour, *op cit*, note 14.

34 J Alderman *Sonic Boom: Napster, P2P and the Future of Music* Fourth Estate, London, 2001.

35 C Shirky 'What P2P is and what isn't', 2000 available at: http://www.openP2P.com/pub/a/P2P/2000/11/24/shirky1-whatisP2P.html.

36 L Gong 'Guest editor's introduction: Peer-to-peer networks in action' *IEEE Internet Computing*, Vol 6, No 1, pp 37–39, 2002.

37 *Ibid*.

38 B Schneier *Applied Cryptography* Wiley, Chichester, 1996.

39 See, eg, F Laurin 'Secret Swedish e-mail can be read by the USA' *Svenska Dagbladet* 18 November 1997.

40 Some countries also had import controls such as France where they existed until 1999.

41 B Gladman 'US government controls on the Microsoft Cryptographic Application Programming Interface' A Paper for the ICE Workshop: International Cryptography Experiment, The Third Workshop, 22 February 1996.

42 *Ibid*.

43 J Kerstetter 'Crypto holes slow export adoption' *PC Week*, 1 June 1998.

44 Microsoft Corporation 'Government Regulation of Cryptography' 1996, available at: http://www.microsoft.com/technet/treeview/default.asp?url = /technet/security/prodtech/mailexch/cryptreg.asp.

45 As an aside, software exports are generally regulated by the Department of Commerce as a dual-use good, but SDKs are regulated by the State Department as a munition.

46 Adapted from Gladman, *op cit*, note 41.

47 *Ibid*.

48 *Ibid*.

49 *Ibid*.

50 B Gladman 'Re: NSA key in Windows' available at UK Crypto Mailing List.