Human Implications of Technology

L. Jean Camp, *Associate Professor, Kennedy School of Government, Harvard University*

Ka-Ping Yee, *University of California, Berkeley*

Abstract

The relationship between technology and society is characterized by feedback. Technological determinists and social determinists perspectives are offer informative but narrow insights into either side of the process. Innovative individuals can generate new technologies that alter the lives, practices, and even ways of thinking in a society. Societies alter technologies as they adopt them, often yielding results far from the hopes or fears of the original designers. Design for values, also called value sensitive design, are methods that explicitly address the human element. sAs the example of usability in security illustrates, a designer who is cognizant of human implications of a design can produce a more effective technological solution.

Overview

L. Jean Camp & Ka-Ping Yee Human implications of technology, *Practical Handbook of Internet Computing* ed. M. P. Singh, CRC Press (New York, NY) Winter 2003.

The human implications of a technology, especially for communications and information technology begin in the design stage. Conversely, human implications of technology are most often considered after the widespread adoption of the technology. Automobiles cause pollution; televisions may cause violence in children.

Social values can be embedded at any stage in the development process: invention, adoption, diffusion, and iterative improvement. A hammer wielded by Habitat for Humanity and a hammer wielded in a violent assault cannot be said to have the same human value at the moment of use; yet the design value of increasing the efficacy of human force applies in both situations.

In the case of the hammer the laws of physics limit the designer. Increasingly the only limit to a designer in the virtual world is one of imagination. Thus designs in search engines, browsers, and even network protocols are created from a previously inconceivably vast range of alternatives.

How important are the choices of the designer? There are two basic schools, one which privileges technical design as the driver of the human condition and one which argues that technologies are the embodiment of social forces beyond the designers' control. After introducing these boundary cases, and identifying the emerging middle, this chapter focuses on a series of studies of particular technical designs. The most considered case is that of security and trust. The chapter closes with a pointer to the significant debates on open or closed code; and the potential of network protocols themselves to embody value-laden choices. The final word is one of caution to the

designer that the single reliable principle of responsible design is full disclosure, as any obfuscation implicitly assumes omnipotence and correctness about the potential human implications of the designers' technical choice.

Technological Determinism

Technological determinism argues that the technologically possible will inevitably be developed and the characteristics of the newly developed technologies will alter society as the technology is adopted [Winner, 1986] [Eisenstein, 1987]. Some find optimism from such a viewpoint, arguing that technology will free us from the human condition [Negroponte, 1995] [Pool, 1983]. Others find such a scenario to be the source of nightmares, arguing that information and communications technologies (ICT) have "laid waste the theories on which schools, families, political parties, religion, nationhood itself" and have created a moral crisis in liberal democracy [Postman, 1996].

Marx has been identified as perhaps the most famous technological determinist in his descriptions of the manner in which the industrial revolution led to the mass exploitation of human labor. Yet technological determinism is not aligned exclusively with any particular political viewpoint.

Technological determinism may be overarching, as with Marx and Winner. In this case, both observed that large complex technologies require large complex organizational systems for their management. The state was required by the structure of large capitalists institutions, which were required by the technologies of the factory and

the railroad. Even government was a function of the organization of capital through the state, as Engels noted, "the proletariat needs the state, not in the interests of freedom but in order to hold down its adversaries, and as soon as it becomes possible to speak of freedom the state as such ceases to exist" [Tucker 1978]. Thus when means and methods of production changed hands the state would no longer be needed. Urban industrialization created that moment in time, at which workers (previously peasants) would be empowered to organize and rise up, overthrowing their oppressors and thereby removing the burden of oppression and government simultaneously.

An echo of this determinist came from the libertarian viewpoint with the publication of the Declaration of Cyberspace. At the peak of the technologically deterministic embrace of the then-emerging digital network, this declaration argued that past technologies had created an invasive oppressive state. Once again the new technologies created under the old state would assure its downfall.

"You claim there are problems among us that you need to solve. You use this claim as an excuse to invade our precincts. Many of these problems don't exist. Where there are real conflicts, where there are wrongs, we will identify them and address them by our means. We are forming our own Social Contract. This governance will arise according to the conditions of our world, not yours. Our world is different." [Barlow, 1996]

Both of these are reductionist political arguments that the state and the society were a function of previous technologies. The changes in technologies would therefore

yield radical changes - in both cases the destruction of the state.

Technological determinism is reductionist. The essential approach is to consider two points in time that differ in the technology available. For example, the stirrup enabled the creation of larger towns by making the care of far-flung fields feasible. Larger towns created certain class and social practices. Therefore determinism would say that the stirrup created social and class re-alignments, and sharpened class distinctions.

In the case of technological determinism of communications the argument is more subtle. Essentially the ICT concept of technological determinism is that media is an extension of the senses in the same way transport is an extension of human locomotion. Thus communications technologies frame the personal and cultural perspective for each participant, as expressed most famously, "The medium is the message." [McLuhan, 1962]

In the case of communications and determinism Marshall McLuhan framed the discourse in terms of the tribal, literate, print, and electronic ages. For McLuhan each technology - the phonetic alphabet, the printing press, and the telegraph - created a new world view. Tribal society was based on stories and magic. Phonetic societies were based on myth and the preservation of cultural heritage. Print societies were based on rational construction and the scientific method. Text reaches for the logical mind; radio and television call out to the most primitive emotional responses. The new hypertext society will be something entirely different, both more reasoned and less rational.

A related debate is geographical determinism [Diamond, 1999] versus cultural determinism[Landes, 1999]. In this argument the distance from the equator and the

availability of natural resources enabled technological development, with the role of culture in innovation being hotly contested. It is agreed that technological development then determined the relative power of the nations of the world.

In McLuhan's view the media technology determined our modern world; with a powerful rational scientific North and a colonized South. In the view of the geographical determinist the geography determined the technology, and technology then determined social outcomes.

Social Determinism

A competing thesis holds that technology is a product of society. This second view is called social construction [Bijker, Hughes, & Pinch, 2001]. Technologies are physical representations of political interests. Social implications are embedded by the stakeholders including inventors and governments, on the basis of their own social values. Some proponents of this view hold that users are the only critical stakeholders. This implies that adoption is innovation and thus the users define technology [Fischer 1992].

As technical determinists look at two points in time and explain all changes as resulting from technology as a social driver, social constructionists look at two points in technological evolution and use social differences as the sole technical driver.

One example of how society drives innovation is in telephony. Both automated switching and low-end telephones were invented for identifiable, explicit social reasons.

Telephones were initially envisioned as business technology and social uses were discouraged, by company policy and advertising campaigns. As long as patents in the United States protected the technology there was no rural market as the technology was also assumed by the Bell Company to be inherently urban. When the patents expired, farmers used their wire fences for telephony and provided part-time low-cost service. The existence of the service, not the quality of the service, was the critical variable. Once intellectual property protections were removed, families adopted phones and social uses of phones came to dominate. Thus telephones were developed that were cheap, less reliable, and of lower quality as opposed to the high-end systems developed by the Bell Company. The design specifications of the telephones were socially determined, but the overall function was technically determined.

In the case of automatic switching, the overall function was socially determined. The goal of an automatic switch was to remove the human switchboard operator. An undertaker, who believed that the telephone operator was connecting the newly bereaved to her brother (the competing town undertaker), invented automated switching [Peirce & Noll 1990]. His design goal was socially determined yet the implementation and specifications for the switch were technical.

Social determinism reflects the obvious fact that identical technologies find different responses in different societies. The printing press created a scientific revolution in Western Europe, but coincided with a decline of science and education in Persian and Arab regions. The printing press had had little effect on the practice of science and

literacy in China by 1600, despite the fact paper and the movable type press had been invented there centuries earlier.

Yet as Barlow illustrates the extreme of technological determinism, social determinism also produces extremes. Robert Fogel received a Nobel Prize in economics for "The Argument for Wagons and Canals" in which he declared that the railroad was of no importance, as canals would have provided the same benefit. The sheer impossibility of building a canal over the Sierra Mountains or across Death Valley were not elements of his thesis, which assumed that canals were feasible within forty miles of any river - including the Rio Grande and various arroyos. A second thesis of Fogel's was that internal combustion would have developed more quickly without the railroad. This thesis ignores the contributions of the railroad engine in the technological innovation and the investments of railroads in combustion innovation. Most importantly this does not acknowledge the dynamics of the emerging engineering profession, as the education and creation of human capital created by the railroad industry were important in internal combustion innovations. Such innovations are less likely to have arisen from those educated to dig canals. This finding requires a reductionist perspective that eliminates all technical distinction from the consideration. In fact an appeal of this paper was the innovation of treating very different technologies as replaceable 'black boxes' while maintaining a mathematically consistent model for their comparison. As is the case with technical determinism, social determinism is reductionist, ignoring technical fundamentals rather than the ubiquitous social environs.

Values in Design

Clearly the social implications of technology cannot be understood be ignoring either the technologies or the social environment. Inclusive perspectives applicable to information and communications technologies have emerged in human-centered design and design for values. These perspectives recognize that technologies can have biases, and attempt to address the values in an ethical manner in the design process. Further, the perspectives explicitly address technologies as biased by the fundamentals of the technology itself as well as through the adoption process.

Social determinism either argues for designers as unimportant cogs to controlling destiny or trivial sideshows to the economic and social questions. Technical determinists view designers as oblivious to their power or omniscient. The mad scientist is a the poster child of dystopian technological determinism.

The design for values or human-centered design school conceives of designers as active participants yet acknowledges the limits of technological determinism. [Friedman and Nissenbaum 2001]. From this design for values perspective, designers offer technological systems and products that bundle functions and values. If the functions are sufficiently useful, the system may be adopted despite undesirable values. If the functions and the values align with the needs of the potential adopters, then widespread adoption is inevitable. While this description sounds clear, determining the values embedded in a system is not trivial.

The examination of values in communications technology is enhanced by past research in the classification of specific values. [Spinello 1996] In privacy there are definitions of private based on system purpose and use (for example the American Code of Fair Information Practice [Federal Trade Commission, 2000]) or only on system use (as with the European Directive on Data Protection [European Union, 1995]). In both cases, there are guidelines intended for users of technology that can be useful for designers of technology. Complicating the lack of examination of inherent and emergent (but theoretically economically predictable) biases is the reality that once adopted, technical standards are difficult to replace.

Biases in communications technologies result from the omission of variables in the design stage (e.g. packet-based networks are survivable and incidentally quality of service is difficult).

Some decisions which may exist in reality as values-determinant are framed by economics only. For example, backward compatibility for nth generation wireless systems is a determinant of cost and therefore of accessibility. Backwards compatibility enables the adoption of obsolete technology for regions that have less capital. In this case a choice on the basis of expense and compatibility can reasonably be said to result in a system which values the marginal first world consumer more or less against the infrastructure needs of the third world consumer. Yet the decision would be made based on the expectations of migration of users of earlier generation technology. Such economic biases are inherent in engineering and design decisions.

In other cases values are embedded through the technical assumptions of the designers, a case well made in a study of universal service and QoS. Similarly privacy risks created in ecommerce are arguably based on the assumption of middle-class designers about the existence of identity-linked accounts. Identity-linked accounts are not available to much of the population. Those without banks or credit cards obviously cannot use identity-linked ecommerce mechanisms. An awareness of privacy would have resulted in technologies usable by those in the cash economy. The plethora of ecash design illustrates that such assumptions can be avoided.

Often design for computer security requires developing mechanisms to allow and refuse trust and thus it may be necessary to embed social assumptions. The issue of human/computer interaction further complicates the design for values system. Any interface must make some assumption about the nature of simplification, and the suitable metaphors for interface (e.g., why does a button make sense rather than a switch or path?). Yet choosing against a simplifying interface is itself a values-laden choice. (See the section on human-computer interaction and security, for example.)

Even when technologists set out to design for a specific value sometimes the result is not always as intended. [Herkert 1999] For example, the Platform for Privacy Preferences has been described by Computer Scientists for Social Responsibility as a mechanism for ensuring that customer data are freely available to merchants; while its designers assert that the goal was customer empowerment. Similarly PICS has been described as a technology for human autonomy [Resnick, 1997] and as "the devil"

[Lessig, 1997] for its ability to enhance the capabilities of censorship regimes worldwide. In both of these cases (not incidentally developed by the World Wide Web Consortium) the disagreement about values is a result of assumptions of the relative power of all the participants in an interaction -- commercial or political.

If the designers' assumptions of fundamental bargaining equality are correct then these are indeed "technologies of freedom" [Pool 1983]. On the other hand the critics of these technologies are evaluating the implementation of these technologies in a world marked by differences in autonomy ranging from those seeking Amnesty International to the clients of the Savoy.

There is no single rule to avoid unwanted implications for values in design, and no single requirement that will embed values into a specific design. However, the following examples are presented in order to provide insights on how values are embedded in specific designs.

Human Implications of the Security Debates

As computer security is inherently the control over the human uses of information, it is a particularly rich area for considering the human implications of technology. The technologically determinant, socially constructed, and design for values models of technological development all have strong parallels in the causes of failures in computer security. Privacy losses result from inherent characteristics of the technology, elements of specific product design, and implementation environments `as well as the

interaction of these three.

Security failures are defined as coding errors, implementation errors, user errors, or so-called human engineering. [Landwher, Bull, McDermott & Choi, 1994]

Security failures are defined as either coding errors or emergent errors, where coding errors are further delineated into either logical flaws in the high level code or simple buffer overruns. The logical, human error, and environment errors correspond loosely to technical determinant, (accidental) social determinant, and iterative embedding of security values in the code.

Coding errors, which can be further delineated into either logical flaws in the high level code or simple buffer overruns, are technologically determinant causes of security failures. The flaw is embedded into the implementation of the technology.

Implementation faults result from unforeseen interactions between multiple programs. This corresponds to the evolutionary perspective, as flaws emerge during adaptation of multiple systems over time. Adoption and use of software in unanticipated technical environments, as opposed to assumptions about the human environment, is the distinguishing factor between this case and the previous one.

User errors are security vulnerabilities that result from the individual failing to interact in the manner prescribed by the system; for example, users selecting weak passwords. These flaws could be addressed by design for values or human-centered design. Assumptions about how people should be (e.g., valuable sources of entropy) as opposed to how they are (organic creatures of habit) are a core cause of these errors.

Alternatively organizational assumptions can create security failures, as models of control of information flow create a systemic requirement to undermine security. Another cause is a lack of knowledge about how people will react to an interface. For example, the SSL lock icon informs the user that there is confidentiality during the connection. Yet the interface has no mechanism to distinguish between confidentiality on the connection and security on the server.

Finally, human engineering means that the attacker obtains the trust of the authorized user and convinces that person to use his or her authorization unwisely. This is a case of a socially determined security flaw. As long as people have autonomy, people will make err. The obvious corollary is that people should be removed from the security loop, and security should be made an unalterable default. It follows that users must be managed and prevented from harming themselves and others on the network. In this case the options of users must be decreased and mechanisms of automated user control are required. Yet this corollary ignores the fallible human involved in the design of security and fails to consider issue of autonomy.

Thus enabling users to be security managers requires educating users to make choices based on valid information. Designs should inform the user and be informed by principles of human-computer interaction.

This section begins with a historical approach to the development of trustworthy systems beginning with the classic approach of building a secure foundation and ending with the recognition that people interact through interfaces, not in raw streams of bits.

Both perspectives are presented, with systems that implement both ideas included in the examples.

Computer security is the creation of a trustworthy system. A secure system is trustworthy in the narrow sense that no data are altered, accessed, or without authorization. Yet such a definition of trustworthy requires perfect authorization policies and practice.

Trustworthy protocols can provide certainty in the face of network failures, memory losses and electronic adversaries. An untrusted electronic commerce system cannot distinguish a failure in a human to comply with implicit assumptions from an attack; in either case transactions are prevented or unauthorized access is allowed. When such failures can be used for profit then certainly such attacks will occur

Trust and security are interdependent. A trusted system that is not compatible with normal human behavior can be subverted using human engineering. Thus the system was indeed trusted, but it was not secure. Trust requires security to provide authentication, integrity and irrefutability. Yet trust is not security; nor does security guarantee trust.

Ideal trustworthy systems inherently recognize the human element in system design. Yet traditional secure systems that focus entirely upon the security of a system without considering its human and social context. [Computer Science and Telecommunications Board, 1999]

Currently there is an active debate expressed in the legal and technical

communities about the fundamental nature of a trustworthy system. [Anderson, 2003][Camp, 2003][Clark and Blumenthal, 2000] This debate can be summed up as follows: Trusted by whom?

Conceptual Approaches to Building Trustworthy Systems

Trusted Computing Base

The fundamental concept of secure computing is the creation of a secure core and the logical construction of provably secure assertions built upon that secure base. The base can be a secure kernel that prevents unauthorized hardware access or secure special-purpose hardware.

The concept of the trusted computing base (TCB) was formalized with the development of the Trusted Computer System Evaluation Criteria (TCSEC) by the Department of Defense. [Department of Defense, 1985] The trusted computing base model sets a series of standards for creating machines, and grades machines from A1 to C2 according to the design and production of the machine. (There is a D rating, which means that the system meets none of the requirements. No company has applied to be certified at the D level.) Each grade, with C2 being the lowest, has increasingly high requirements for security beginning with the existence of discretionary access control, meaning that a user can set constraints on the files. The C level also requires auditing and authentication. The higher grade, B, requires that users be able to set security constraints on their own resources and be unable to alter the security constraints on

documents owned by others. This is called mandatory access control.

The TCB model as implement in the Trusted Computer System Evaluation Criteria is ideal for special-purpose hardware systems. The TCB model becomes decreasingly applicable as the computing device becomes increasingly general purpose. A general purpose machine, by definition, can be altered to implement different functions for different purposes. The TCSEC model addresses this by making functionality and implementation distinct. Yet logging requirements, concepts of document ownership, and the appropriate level of hardening in the software change over time and with altered requirements.

While multiple systems have sought and obtained C level certification under the TCSEC, the certification is not widely used in the commercial sector. Military emphasis on information control differs from civilian priorities in three fundamental ways. First, in military systems it is better that information be destroyed than exposed. In civilian systems the reverse is true: bank records, medical records, and other personally identifiable information are private but critical. Better a public medical decision than a flawed diagnosis based on faulty information.

Second, the military is not sensitive to security costs. Security is the reason for the existence of the military, for others it is an added cost to doing business.

Third, the Department of Defense in unique in its interactions with its employees and members. The Department of Defense and its individual participants are uniquely tightly aligned. There is no issue of trust between a soldier and commander. If the

Department determines its policies then the computer can implement those policies. Civilians, businesses, families, and volunteer organizations obviously have very different organizational dynamics.

One goal of the TCB is to allow a centralize authority to regulate information system use and access. Thus the Trusted Computing Based may be trusted by someone other than the user to report upon the user. In a defense context, given the critical nature of the information, users are often under surveillance to prevent information leakage or espionage. In contrast, a home user may want to be secure against the Internet Service Provider as well as remote malicious users.

Microsoft's Next Generation Secure Computing Platform is build on the trusted computing base paradigm.

Next Generation Secure Computing Platform

Formerly known as Palladium, then the Trusted Computing Base, then the Next Generation Secure Computing Platform the topic of this section is now called Trusted Computing (TC). Regardless of the name, this is a hotly contested security design grounded in the work of Microsoft. TC requires a secure coprocessor that does not rely on the larger operating system for its calculations. In theory the TC can even prevent the operating system from booting if the lowest level system initiation (the BIOS, or basic input/output system which loads the operating system) is determined by the TC to be insecure. The TC must include at least storage for previously calculated values, and the capacity for fast cryptographic operations.

A primary function of the TC is to enable owners of digital content to control that content in digital form. [Anderson, 2003] TC binds a particular data object to a particular bit of software. Owners of high value commodity content have observed the widespread sharing of audio content enabled by overlay networks, increased bandwidth, and effective compression. By implementing players in 'trusted' mode video and audio players can prevent copying, and enforce arbitrary licensing requirements by allowing only trusted players. (In this case the players are trusted by the content owners, not the computer user.) This is not limited to multimedia formats, as TC can be used for arbitrary document management. For example, the Adobe eBook had encryption to prevent the copying of the book from one device to another, prohibit audio output of text books, and to enforce expiration dates on the content.[Adobe, 2002] The Advanced eBook Processor enabled reading as well as copying, with copying inherently preventing deletion at the end of the licensed term of use. Yet with TCB an Adobe eBook could only be played with an Adobe eBook player, so the existence of the Advanced eBook Processor would not threaten the Abode licensing terms.

In physical markets encryption has been used to bind future purchases to past purchases, in particular to require consumers of a durable good to purchase related supplies or services from the manufacturer. Encryption protects ink cartridges for printers to prevent the use of third-party printer cartridge use. Patents, trade secrets and encryption link video games to consoles. Encryption connects batteries to cellular phones, again to prevent third-party manufacturers from providing components.

Encryption enables automobile manufacturers to prevent mechanics from understanding diagnostic codes and, if all is working smoothly, from turning off warning lights after an evaluation. [Prickler, 2002]

Given the history of Microsoft and its use of tying, the power of the TC for enforcing consumer choices is worth consideration. It is because of the potential to limit the actions of users that the Free Software Foundation refers to the effort as "Treacherous Computing" [Stallman, 2002]. Microsoft was found to be guilty of monopolistic anti-competitive actions in binding the Explorer browser to the operating system. Currently Microsoft is facing competition from open code, including StarOffice and GNU/Linux. Were the TC implemented at the original equipment manufacturer it could be impossible to remove the Microsoft operating system and replace it with Linux.

Microsoft also holds a monopoly position in desktop publishing, spreadsheet, and presentation software with property interests in the corresponding MS word processing (doc), Excel spreadsheet (xls), and PowerPoint Presentation (ppt) document formats. By combining encryption with storage the TC would enforce lock-in. That is, the TC could refuse to decrypt documents to any application except one that could attest to being a legal copy of Microsoft application. Were competitors to reverse engineer the encryption to enable reading the documents their action would be felonious under the Digital Millennium Copyright Act.

Initially Palladium explicitly included the ability to disable the hardware and software components, so that the machine could run in "untrusted" mode. The recent

integration of document control features in the MS Office suite requires that TC be enabled for any manipulation of the MSOffice documents (reading, saving, altering).

The TC centralizes power and trust. The centralization of trust is a technical decision. Using IBM implementations of TC it is possible to load Linux. IBM has made the driver code for the TCPA compatible chip (which they distinguish from Palladium) available over the Internet with an open license, and offer the product with Linux.

The TC makes it possible to remove final (or root) authority from the machine owner, or to allow the owner to control her own machine more effectively. Thus the design leverages and concentrates power in a particular market and legal environment.

The design for values perspective argues the TC is valuable only if it provides root access and final authority to the end user. Yet TC is built in order to facilitate removal of owner control. TC offers two parties authorization -- an operator who is assumed to have physical access to the machine and an owner with remote access. The remote owner is specifically enabled in its ability to limit the software run by the owner. A design goal of TC is that the operator cannot reject alterations made by the owner, in that the typical operator cannot return the machine to a previous state after an owner's update.

In short, TC is designed to remove control from the end user (or operator) and place that control with a remote owner. If the operator is also the owner, TC has the potential to increase user autonomy by increasing system security. If the owner is in opposition to the operator then the operator has lost autonomy by virtue of the increased

security. The machine is more secure and less trustworthy from the perspective of the owner.

Human-Centered Trusted Systems Design

Technical systems, as explained above, embody assumptions about human responses [Camp, McGrath, & Nissenbaum, 2001]. That humans are a bad source of randomness is well documented, and the problems of "social engineering" are well known [Anderson, 2002]. Yet the consideration of human behavior has not been included in classic axiomatic tests [Aslam, Krsul, & Spafford, 1996; Anderson, 1994].

For example, designers of secure systems often make assumptions about the moral trust of humans, which is a psychological state, and strategic trust of machines [Shneiderman, 2000][Friedman, Kahn, & Howe, 2000]. Yet user differentiation between technical failures and purposeful human acts of malfeasance has never been tested. Despite the fact that the current software engineering process fails to create trustworthy software [Viego, Kohno, & Potter, 2001] much work on trust-based systems assumes only purposeful betrayals or simply declares that the user should differentiate [Friedman, Kahn, & Howe, 2000].

The inclusion of human factors as a key concern in the design of secure systems is a significant move forward in human-centered design. Human-centered design attempts to empower users to make rational trust decisions by offer information in an effective manner. "Psychological acceptability" was recognized a security design principle over a quarter century ago [Saltzer & Schroeder, 1975], and users and user behavior are

commonly cited as the "weak link" in computer security. Passwords and other forms of authentication are among the more obvious ways that security features appear as part of the human-computer interface. But the relevance of computer-human interaction to computer security extends far beyond the authentication problem, because the expectations of humans are an essential part of the definition of security. For example, Garfinkel and Spafford suggested the definition: "A computer is secure if you can depend on it and its software to behave as you expect" [1996]. Since goals and expectations vary from situation to situation and change over time in the real world, a practical approach to computer security should also take into account how those expectations are expressed, interpreted, and upheld.

Although both the security and usability communities each have a long history of research extending back to the 1960s, only more recently have there been formal investigations into the interaction between these two sets of concerns. Some usability studies of security systems were conducted as early as 1989 [Karat, 1989; Mosteller & Ballas, 1989]. However, with the advent of home networking in the late 1990s, the study of computer-mediated trust has significantly expanded.

General Challenges

It is fairly well known that usability problems can render security systems ineffective or even motivate users to compromise security measures. While HCI principles and studies can help to inform the design of usable security systems, merely applying established HCI techniques to design more powerful, convenient, or lucid

interfaces is not sufficient to solve the problem - the challenge of usable security is uniquely difficult. There are at least six special characteristics of the usable security problem that differentiate it from the problem of usability in general [Whitten, 1999; Sasse, 2003]:

1. *The barn door property*: Once secret information has been left unprotected, even for a short time, there is no way to be sure that an attacker has not already obtained it. Once access has been inadvertently allowed, even for a short time, there is no way to be sure that an attacker has not already abused that access.
2. *The weakest link property*: When designing user interfaces in most contexts, a deficiency in one area of an interface does not compromise the entire interface. However, a security context is less forgiving. The security of a networked computer is only as strong as its weakest component, so special care needs to be taken to avoid dangerous mistakes.
3. *The unmotivated user property*: Security is usually a secondary goal, not the primary purpose for which people use their computers. This can lead users to ignore security concerns or even subvert them when security tasks appear to interfere with the achievement of their primary goal.
4. *The abstraction property*: Security policies are systems of abstract rules, which may be alien and unintuitive to typical computer users. The consequences of making a small change to a policy may be far-reaching

and non-obvious.

5. *The lack of feedback property*: Clear and informative user feedback is necessary in order to prevent dangerous errors, but security configurations are usually complex and difficult to summarize.

6. *The conflicting interest property*: Security, by its very nature, deals with conflicting interests - such as the interests of the user against the interests of an attacker, or the interests of a company against the interests of its own employees.

HCI research typically aims to optimize interfaces to meet the needs of a single user or a set of co-operating users, and is ill-equipped to handle the possibility of active adversaries. Because computer security involves human beings, their motivations, and conflicts among different groups of people, security is a complex socio-technical system.

Authentication

Since user authentication is a very commonly encountered task and a highly visible part of computer security, much of the attention in usable security research has been devoted to this problem. The most common authentication technique, of course, is the password. Yet password authentication mechanisms fail to acknowledge even well-known HCI constraints and design principles [Sasse et al., 2001]. Cognition research has established that human memory decays over time, that non-meaningful items are more difficult to recall than meaningful items, that unaided recall is more difficult than cued recall, and that similar items in memory compete and interfere with each other during

retrieval. Password authentication requires perfect unaided recall of non-meaningful items. Furthermore, many users have a proliferation of passwords for various systems or have periodically changing passwords, which forces them to select the correct password from a set of several remembered passwords. Consequently, people often forget their passwords, and rely on secondary mechanisms to deal with forgotten passwords.

One solution is to provide a way to recover a forgotten password or to reset the password to a randomly generated string. The user authorizes recovery or reset by demonstrating knowledge of a previously registered secret or by calling a helpdesk. There are many design choices to make when providing a challenge-based recovery mechanism [Just, 2003]. Another common user response to the problem of forgetting passwords is to write down passwords, or to choose simpler passwords that are easier to remember, thereby weakening the security of the system. One study of 14,000 UNIX passwords found that nearly 25% of all the passwords were found by trying variations on usernames, personal information, and words from a dictionary of less than 63,000 carefully selected words [Klein, 1990].

In response to all the shortcomings of user-selected string passwords, several password alternatives have been proposed. Jermyn et al. [Jermyn et al., 1999] have examined the possibility of using hand-drawn designs as passwords. Others have looked at recognition-based techniques, where users are presented with a set of options and are asked to select the correct one, rather than performing unaided recall. Brostoff & Sasse [2000] studied the effectiveness of images of human faces in this manner, and Dhamija &

Perrig [2000] studied the use of abstract computer-generated images. In contexts where it is feasible to use additional hardware, other solutions are possible. Users can carry smart cards that generate password tokens, or that produce responses to challenges from a server. Paul et al. [2003] describe a technique called "visual cryptography" in which the user overlays a uniquely coded transparency over the computer screen to decrypt a graphical challenge, thereby proving possession of the transparency.

There is considerable interest in using biometrics for user authentication. A variety of measurable features can be used, such as fingerprints, voiceprints, hand geometry, faces, or iris scans. Each of the various methods has its own advantages and disadvantages. Biometrics offer the potential for users to authenticate without having to remember any secrets or carry tokens. However, biometrics have the fundamental drawback that they cannot be reissued. A biometric is a password that can never be changed. Once compromised, a biometric is compromised forever. Biometrics raise significant concerns about the creation of centralized databases of biometric information, as a biometric (unless hashed) creates a universal identifier. Biometrics also have value implications in that biometric systems most often fail for minorities [Woodward, Webb & Newton, 2003]. Biometrics present class issues as well; for example, biometric records for recipients of government aid are already stored in the clear in California. The storage of raw biometric data makes compromise trivial and thus security uncertain.

User Perceptions and Trust

User perceptions of security systems are crucial to their success in two different

ways. First, the perceived level of reliability or trustworthiness of a system can affect the decision of whether to use the system at all; second, the perceived level of security or risk associated with various choices can affect the user's choice of actions. Studies [Cheskin, 1999; Turner et al., 2001] have provided considerable evidence that user perception of security on e-commerce Web sites is primarily a function of visual presentation, brand reputation, and third-party recommendations. Although sufficiently knowledgeable experts could obtain technical information about a site's security, for ordinary consumers, "feelings about a site's security were for the most part not influenced by the site's visible use of security technology" [Turner et al., 2001].

With regard to the question of establishing trust, however, perceived security is not the whole story. Fogg conducted a large study of over 1400 people to find out what factors contributed to a Web site's credibility [2001]. The most significant factors were those related to "real-world feel" (conveying the real-world nature of the organization, such as by providing a physical address and showing employee photographs), "ease of use," and "expertise" (displaying credentials and listing references). There is always an effect of presenting photographs of people on Web sites, but the effect is not always positive [Riegelsberger, 2003].

In order to make properly informed decisions, users must be aware of the potential risks and benefits of their choices. It is clear that much more work is needed in this area. For example, a recent study showed that many users, even those from a high-technology community, had an inaccurate understanding of the meaning of a secure

connection in their Web browser, and frequently evaluated connections as secure when they were not or vice versa [Friedman et al., 2002].

A recent ethnographic study [Dourish, 2003] investigated users' mental models of computer security. The study revealed that users tend to perceive unsolicited e-mail, unauthorized access, and computer viruses as aspects of the same problem, and envision security as a barrier for keeping out these unwanted things. The study participants blended privacy concerns into the discussion, perceiving and handling marketers as threats in much the same way as hackers. However, there seemed to be an "overwhelming sense of futility" in people's encounters with technology. The perception that there will always be cleverer adversaries and new threats leads people to talk of security in terms of perpetual vigilance.

In order to make properly designed systems, designers must be aware of human practices with respect to trust. Studies of trust in philosophy and social science argue that humans trust readily, and in fact have a need to trust. Further, humans implement trust by aggregating rather than differentiating. That is, humans sort entities into trustworthy and untrustworthy groups. Thus, when people become users of computers, they may aggregate all computers into the class of computers, and thus become increasingly trusting rather than increasingly differentiating over time [Sproull & Kiesler, 1992]. Finally, when using computers humans may or may not differentiate between malicious behavior and technical incompetence. Spam is clearly malicious while privacy violations may be a result of an inability to secure a website or a database. Competence in web

design may indicate a general technical competence, thus mitigating concerns about competence. However, competence in web design may also indicate efficacy in obtaining user's trust for malicious purposes. Only the ability to discern the technical actions and understand the implications can provide users with the ability to manage trust effectively on the network.

Interaction Design

A number of studies have shown the potential for problems in interaction design to seriously undermine security mechanisms. Whitten [1999] demonstrated that design problems with PGP made it very difficult for even technically knowledgeable users to safely use e-mail encryption; a study by Good [2003] identifies problems in the user interface for KaZaA that can lead to users unknowingly exposing sensitive files on their computer. Carl Ellison has suggested that each mouse click required to use encryption will cut the base of users in half. [Ellison 2002]

Results of such studies and personal experiences with security systems have led researchers to propose a variety of recommendations for interaction design in secure systems. Yee has proposed ten principles for user interaction design in secure systems [2002]. At a higher level are recommendations to apply established HCI techniques to the design process itself. Karat described the benefits of applying rapid prototyping techniques to enable an iterative process involving several rounds of field tests and design improvements [1989]. Zurko and Simon [1996] suggest applying user-centered design to security - that is, beginning with user needs as a primary motivator when

defining the security model, interface, or features of a system. Grinter and Smetters suggested beginning the design process with a user-centered threat model and a determination of the user's security-related expectations [2003]. Techniques such as contextual design [Wixon, 1990] and discount usability testing [Nielsen, 1989] are also applicable to interaction design for secure systems.

Some have suggested that the best way to prevent users from making incorrect security decisions is to avoid involving users in security at all. Others argue that only the user really knows what they want to do, and knowledge of the user's primary goal is essential for determining the correct security action. It is clear that forcing users to perform security tasks irrelevant to their main purpose is likely to bring about the perception that security interferes with real work. Yee has suggested the *principle of the path of least resistance*, which recommends that the most natural way to perform a task should also be the safest way [2002]. Sasse highlighted the importance of designing security as an integral part of the system to support the user's particular work activity [2003]. Grinter and Smetters have proposed a design principle called *implicit security*, in which the system infers the security-related operations necessary to accomplish the user's primary task in a safe fashion [2003]. The perspectives are similar, and all recognize the necessity of harmonizing security and usability goals rather than pitting them against each other.

In order for users to manage a computer system effectively, there must be a communication channel between the user and the system that is safe in both directions.

The channel should protect against masquerading by attackers pretending to be authorized users, and protect users from being fooled by attackers spoofing messages from the system. This design goal was identified by Saltzer and Schroeder as the *trusted path* [1975]. Managing complexity is a key challenge for user interfaces in secure systems. Grinter [2003] and Yee [2002] have identified the need to make security state visible to the user so that the user can be adequately informed about the risks, benefits, and consequences of decisions. However, a literal representation of all security relationships would be overwhelming. Cranor [2003] applied three strategies to deal with this problem in designing a policy configuration interface: (a) reducing the level of detail in the policy specification; (b) replacing jargon with less formal wording; and (c) providing the option to use pre-packaged bundles of settings. Whitten [2003] has suggested a technique called *safe staging*, in which users are guided through a sequence of stages of system use to increase understanding and avoid errors. Earlier stages offer simpler, more conservative policy options for maintaining security; then as the user moves to later stages, she gains progressively greater flexibility to manipulate security policy while receiving guidance on potential new risks and benefits.

Whitten's usability study of PGP [1999] showed strong evidence that designing a user interface according to traditional interface design goals alone was not sufficient to achieve usable security. Whitten recommends that usable security applications should not only be easy to use, but also should teach users about security and grow in sophistication as the user demonstrates increased understanding. Ackerman and Cranor

[1999] proposed "critics" - intelligent agents that offer advice or warnings to users, but do not take action on the user's behalf. Such agents could inform users of non-obvious consequences of their actions, or warn when a user's decision seems unusual compared to decisions previously made in similar situations.

Identity Examples

Systems for identity management are inherently social systems. These systems implement controls that place risk, control data flows, and implement authentication. The strong link between control and computer security, and between identity and privacy, make these ideal examples for considering social implications of technology.

PKI

Public signatures create bindings between identifiable cryptographic keys and specific (signed) documents. Public key infrastructures serve to link knowledge of a particular key to particular attribute. Usually that attribute is a name, but there are significant problems with using a name as a unique identifier. [Ellison & Camp, 2003]

The phrase 'public key infrastructure' has come to refer to a hierarchy of authority. There is a single root or a set of root keys. The root keys are used to sign documents (usually quite short, called certificates) that attest to the binding between a key and an attribute. Since that attribute is so often identity, the remainder of the section assumes it is indeed identity.

Thus each binding between a key and identity is based on a cryptography verification from some other, higher-level key. At the base is a key that is self-certified.

Standard PKI implements a hierarchy with the assumption of a single point from which all authority and trust emanates. The owner of the root key is a certificate authority.

The current public key infrastructure market and browser implementation (with default acceptable roots) implement a concentration of trust. Matt Blaze argues that the SSL protects you from any institution that refuses to give Verisign money [2003]. The cryptographer Blaze arguably has an accurate assessment, as the purchaser of the cryptographic verification determines Verisign's level of investigation into the true identity of the certificate holder.

Public key infrastructures centralize trust by creating a single entity that signs and validates others. The centralization of trust is further implemented by the selection of a set of keys which are trusted by default by a market that is a duopoly or monopoly.

PGP

Confidentiality in communications was the primary design goal of Pretty Good Privacy. PGP was conceived as a privacy enhancing technology as opposed a technology for data protection. [Garfinkel, 1999] This fundamental distinction in design arguable explains the distinct trust models in the two technologies.

Pretty Good Privacy allows any person to assert an identity and public key binding. It is then the responsibility of the user to prove the value of that binding to another. In order to prove the binding the user presents the key and the associated identity claim to others. Other individuals who are willing to assert that the key/identity binding is correct sign the binding with their own public keys. This creates a network of

signatures, where any user may or may not have a trusted connection.

PGP utilizes social networks. If PKI can be said to model an authoritarian approach, PGP is libertarian. PKI has roots that are privileged by default. Each PGP user selects parties that are trusted not only for their assertions about their own binding of key and identity but also for their judgment in verifying the linkage of others. Those who have trusted attestations are called introducers. These introducers serve as linking points between the social networks formed by the original user and other social networks.

PGP has monotonically increasing trust. When an introducer is selected that introducer remains trusted over an infinite period of time unless manually removed. If an introducer is manually removed all the introduced parties remain trusted, as the paths to a trusted entity are not recorded after introduction.

PGP increases trust as an increasing number of introducers know another entity. PGP does not decrease trust if one introducer declares a lack of knowledge regardless of the properties of the social network.

PGP was designed to enable secure email. Secure email provides both integrity of the content and authentication of the sender. PGP enables confidential email by providing the endpoints with the capacity to encrypt.

PGP places final trust in the hands of the user, and allows to user to implement her own social network by creating introducers. The same underlying cryptography is used by PGP and PKI but the values choices embedded are distinct.

Data Protection versus Privacy

L. Jean Camp & Ka-Ping Yee Human implications of technology, *Practical Handbook of Internet Computing* ed. M. P. Singh, CRC Press (New York, NY) Winter 2003.

The focus on human implications of design have focused on trust and trusted systems. However, privacy as well as security is an element of trust.

Data surveillance, privacy violations, or abuse of data (depending on the jurisdiction and action) can be both ubiquitous and transparent to the user. Computers may transmit information without the users' knowledge; and collection, compilation and analysis of data is tremendously simplified by the use of networked information systems. Because of these facts, the balance between consumers and citizens who use services and those that offer digital services cannot be maintained by simply moving services on-line.

A consideration of social implications of technology should include the dominant privacy technologies. The two most widely used (and implemented) privacy enhancing technologies are the anonymizer and P3P. The anonymizer implements privacy while P3P implements a data protection regime.

Data protection and privacy have more commonalities than differences. The differences have philosophical and as well as technical design implications.

Technical decisions determine the party most capable of preventing a loss of security, policy decisions determine can motivate those most capable. Current policy does not reflect the technical reality - end users are least technically capable and most legally responsible for data loss. For the vast majority of users on the Internet there are programs beyond their understanding and protocols foreign to their experience. Users of the Internet know that their information is sometimes transmitted across the globe. Yet there is no way for any but the most technically savvy consumers to determine the data

leakage that result from Internet use.

There is a comprehensive and developed argument for data protection. The privacy argument for data protection is essentially that when the data are protected privacy is inherently addressed. One argument against privacy is that it lacks a comprehensive, consistent underlying theory. There are competing theories [Camp, 2003b] [Trublow 1991] [Kennedy 1995] yet the existence of multiple, complete yet disjoint theories does illustrate the point that there is limited agreement. Data protection regimes offer to address problems of privacy via prohibition of data reuse and constraints on compilation. By focusing on practical data standards, data protection sidesteps difficult question of autonomy and democracy that are inherent in privacy. Data protection standards constrain the use of personally identifiable information in different dimensions according to the context of the compilation of the information and the information itself.

Unlike data protection, privacy has a direct technical implementation: anonymity.

For both privacy and data protection, one significant technical requirement is enabling users to make informed choices. Given the role of security technology in enhancing privacy human-computer design for security can enhance both privacy and data protection.

Anonymity provides privacy protection by preventing data from being personally identifiable. Anonymity provides the strongest protection possible for privacy, and anonymity provides uniform protection for privacy across all circumstances. Anonymity

is of limited in value in situations where there is a need to link repeated transaction. In such cases pseudonyms are needed. Pseudonyms can have no link to other roles or true identity; for example a pseudonym may be a name used in an imagined community such as a role-playing game. Pseudonyms allow for personalization without privacy violations. Repeated use of a pseudonym in multiple transactions with the same entities leaks little information. Use of a pseudonym in multiple contexts (for example, with multiple companies) causes the pseudonym to converge with the identity of the user.

Privacy enhancing technologies include technologies for obscuring the source and destination of a message (onion routing) and preventing information leakage while browsing (the anonymizer).

Onion routing encrypts messages per hop suing an overlay network of routers with public keys. At each router, the message provides the address of the next router and a message encrypted with the public key of the next router. Thus each router knows the source of the message and the next hop, but not the original source nor the final destination. However, the router records could be combined to trace a message across the network. Yet even with the combined records of the routers the confidentiality of the message would remain.

The anonymizer is a widely used privacy-enhancing proxy.

The anonymizer implements privacy by functioning as an intermediary so that direct connections between the server and the browser are prevented. The anonymizer detects web bugs. Web bugs are 1x1 invisible images embedded into pages to allow

entities other than the serving page to track usage. Since web bugs are placed by a server other than the one perceived by the user, web bugs allow for placement of cookies from originating server. This subverts user attempts to limit the use of third-party cookies. Note that browsers have a setting the allows users to reject cookies from any server other than one providing the page, so-called third party cookies. Web bugs enable placement of third party cookies. The anonymizer also limits java script and prevents the use of ftp calls to obtain user email addresses. The anonymizer cannot be used in conjunction with purchasing, it is limited to browsing.

In contrast, data protection encourages protections based on policy and practice.

The Platform for Privacy Preferences is a technology that implements the data protection approach. [Cranor & Reagle 1998]. The Platform for Privacy preferences was designed to create a technical solution to the problem of data sharing. Ironically, of all privacy-enhancing technologies, it depends on regulatory enforcement of contracts as opposed to offering technical guarantees. [Hochheiser 2003]

The Platform for Privacy Preferences includes a schema (or language) for expressing privacy preferences. P3P allows the user to select a set of possible privacy policies by selecting a number <1,10> on a sliding scale. P3P also has a mechanism for a service provider to express its privacy practices. If there is agreement between the server policy and user preference then user data are transmitted to the server. Otherwise no data are sent.

P3P also includes profiles where the user enters data. The inclusion of profiles

has been a significant source of criticism as it removes from the user the ability to provide incorrect data. By including profiles, P3P removed from the user a common defense against data re-use (obfuscation, lying or misdirection) by automating the transmission of correct data. P3P therefore had an enforcement role with respect to user data -either the user was consistently dishonest or consistently honest. P3P had no corresponding enforcement mechanism for the server. The server attests to its own privacy policy and the protocol assumes the server implements its own policy. The most recent version of P3P removes profiles; however the MS Explorer implementation still maintains profiles.

Network Protocols as Social Systems

Network protocols for reservation of system resources in order to assure quality of service is a technical area of research. However, even quality of service designs have significant impact on the economics, and therefore the social results, of such systems. There is no more clear example of politics in design than the change in the design of Cisco routers to simplify so-called "digital wiretaps." Wiretaps refer to aural surveillance of a particular set of twisted pair lines coming from local switching office [IEEE 1997]. This simple model was often abused by Federal authorities in the Untied States, and use of aural surveillance against dissidents, activists and criminals has been widespread across the globe [Diffee & Landau 1997]. "Digital telephony" is a phrase used by law enforcement to map the concept of wiretaps to the idea of data surveillance of an

individual on the network. If the observation of digital activity is conceived as a simple digital mapping then the risks can be argued as the same. Yet the ease of compilation and correlation of digital information argues that there is not a direct mapping.

Cisco implemented a feature for automatically duplicating traffic from one IP address to a distinct location. Thus Cisco implemented a feature desired by law-abiding consumers of the product and by law enforcement. However, Cisco implemented no controls or reporting on this feature. Therefore Cisco altered the balance of power between those under surveillance and those implementing surveillance by lowering the work factor for the latter. Furthermore, the invisibility of surveillance at the router level further empowers those who implement surveillance. The ability to distinguish the flow from one machine across the network and to duplicate that flow for purposes of surveillance is now hard-wired into the routers. Yet the oversight necessary to prevent abuse of that feature is not an element of the design; not even one that must be disabled. An alternative configuration would require selection of a default email address to which notifications of all active taps would be sent. The email could be set according to the jurisdiction of the purchaser; thus implementing oversight.

A more subtle question is the interaction of quality of service mechanisms and universal service. The experience of the telephone (described in a previous section) illustrates how high quality service may limit the range of available service. Universal service may require a high degree of service sharing and certainly requires an easy to understand pricing method. Ubiquitous service reservation, and the resulting premium

pricing, can undermine the potential of best effort service to provide always on connections at a flat price. [Camp and Tsang 2002]

Open Versus Closed Code

There exists a strong debate on how the availability of code alters the society as digital systems are adopted. The initiator of this dialogue was Richard Stallman, who saw the corporate closing of code. [Stallman, 1984] Other technical pioneers contributed both to the code base and the theory of free and open code. [Oram 1999][Raymond 1999] Legal pioneers [Branscomb, 1984] clearly saw the problem of closing information by the failure of property regimes to match the economic reality of digital networked information. By 2000 [Lessig, 1999] there was widespread concern about the social implications of the market for code. Code can be distributed in a number of forms that range from completely open to completely closed. Code is governed by licenses as well as law; yet the law is sufficiently slow to adapt that graduations of and innovations in openness are provided by licenses.

Computer code exists along a continuum. At one end is source code. Source code is optimized for human readability and malleability. Source code is high level code, meaning that it is several degrees removed from the physical or hardware level. An example of inherently human readable code is mark-up languages. There are technical means to prohibit the trivially easy reading and viewing of a document source, and methods for writing increasingly obtuse source code are proliferating. For example,

popular Web-authoring documents use unnecessary Java calls to confuse the reading. At the most extreme is converting html-based Web pages to Shockwave formats which cannot be read. Markup languages and scripting languages such as JavaScript and CGI scripts are designed to be readable. Such scripts are read (thus the name) and then the listed commands are played in the order received.

Between the highest level and the physical addresses required by the machine, there is assembly language. Assembly is the original coding language. Grace Hopper (who found the original computer bug, a moth in the machine at Harvard in 1945) implemented programs in assembly. Assembly requires that humans translate program into the binary language that machines understand. For example, a adding two numbers in assembly takes many lines of code. The computer must be instructed to read the first number, but by bit, and store it in an appropriate location. Then the computer must read the second number. Then the numbers must be placed at the input to the arithmetic logic unit, then added, and the result placed in an output register. Grace Hopper invented the breakthrough of the assembler, the forerunner to the compiler.

The earliest code was all binary; of course, and thus clearly the most basic binary codes can be read. In these early binary the commands were implemented by women who physically linked nodes to create the binary "1" of the commands. For each mathematician creating a code there existed a large machine and a score of women to implement it the commands by connecting two relays, thus programming a "1".

Current programmers are vastly more complex than those implemented in hand-

wired binary. The breaking of the Enigma machine was a vast enterprise, with Alan Turing's achievement honored by statue in the United Kingdom. Today the same endeavor is an advanced undergraduate homework assignment. Thus machine (sometimes called binary) code for today programs is unreadable.

It is the ability to read code that makes it open or closed. Code that can be read can be evaluated by the user or the representative of the user. Code that is closed and cannot be read requires trust in the producer of the code. Returning to social forces that influence technology, this is particularly problematic. A user wants a secure machine. The producer of commercial code has an incentive to create code as fast as possible, meaning that security is not a priority. The user wants control over his or her personal information. The producer of commercial code may want information about the user, particularly to enforce intellectual property rights [Anderson, 2003] or to implement price discrimination [Odlyzko, 2003].

The ability to read code grants the practical ability to modify it. Of course, the existence of closed code does not prevent modifications. This can be seen most clearly in the modifications of the Content Scrambling System. The decryption of the Content Scrambling System enabled users to watch digital video disks from any region on any operating system. CSS implements the marketing plans, specifically regional price discrimination, that is the traditional and profitable practice of the owners of mass-produced high-value video content.

Open code encourages innovation by the highly distributed end users by

optimizing opportunities for innovation. Closed code encourages innovation by increasing the rewards to the fewer, centralized innovators. Thus open and closed code implement different visions of innovation in society.

Open code offers transparency. Closed code is not transparent. If code is law, then the ability to view and understand law is the essence of freedom. [Lessig 1999] [Stallman 1984] [Syme & Camp 2001]. The inability to examine law is a mark of a totalitarian state [Solzhenitsyn, 1975].

Conclusions

Can the assumption of values be prevented by the application of superior engineering? Or are assumptions about values and humans an integral part of the problem-solving process? Arguably both cases exist in communications and information technologies. These two cases cannot be systematically and cleanly distinguished so that the designer can know when the guidance of philosophy or social sciences is most needed. When is design political? One argument is that politics inevitably intrudes when there are assumptions about trust and power embedded into the design. Yet trust assumptions may be as subtle as reservation of router resources or as obvious as closed code.

Technologies embed changes in power relationships by increasing the leverage of applied force. Yet the social implications of amplification of one voice or one force cannot be consistently or reliably predicted.

Designers who turn to the study of computer ethics find the field not yet mature. There are some who study ethics that argue that computers create no new ethical problems, but rather create new instantiations of previous ethical problems [Johnson 2001]. Others argue that digital networked information creates new classes or cases of ethical conundrums [Walter 1996] [Moor 1985]. Ethicists from all perspectives worked on the relevant professional codes. Thus the professional can be guided by the ACM/IEEE- CS Software Engineering Code of Ethics and Professional Practice.

Integrity and transparency are the highest calling. No engineer should implement undocumented features, and all designers should document their technical choices.

The most risk averse principle of design scientist may be 'do no harm'; however, following that principle may result in inaction. Arguably inaction in the beginning of a technological revolution is the least ethical choice of all, as it denies society the opportunity to make any choices, however technically framed.

Additional Resources

The IEEE Society on Social Implications of Technology

http://radburn.rutgers.edu/andrews/projects/ssit/default.htm

ACM SIGCAS: Special Interest Group on Computers and Society

http://www.acm.org/sigcas/

An extended bibliography on technology and society, developed by a reference librarian:

http://www.an.psu.edu/library/guides/sts151s/stsbib.html

A listing of technology and society groups, and electronic civil liberties organizations: http://www.ljean.org/eciv.html

References

Michael Ackerman and Lorrie Cranor. Privacy critics: UI components to safeguard users' privacy. *Proceedings of CHI 1999*.

Adobe Corporation. Adobe eBook FAQ, 2002. http://www.adobe.com/support/ebookrdrfaq.html

Ellen Alderman and Caroline Kennedy, The Right to Privacy. Alfred A Knopf, 1995.

Ross Anderson. Why cryptosystems fail. *Communications of the ACM*, 37(11): 32-40, November 1994.

Ross Anderson. Cryptography and Competition Policy - Issues with Trusted Computing. 2nd Annual Workshop on Economics and Information Security (May 29-30, 2003, Robert H. Smith School of Business, University of Maryland).

Ross Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley & Sons, 2002.

Taimur Aslam , Ivan Krsul , and Eugene Spafford, A Taxonomy of Security Vulnerabilities. *Proceedings of the 19th National Information Systems Security Conference* (October 6, 1996, Baltimore, Maryland), 551-560.

John Barlow. A Declaration of Independence of Cyberspace. http://www.eff.org/~barlow/Declaration-Final.html, 1996 (last viewed September,

2003).

Wiebe Bijker , Thomas P. Hughes , Trevor Pinch *The Social Construction of Technological Systems*. MIT Press, 2001.

Matt Blaze. Quotes, August 31, 2003. http://world.std.com/~cme/html/quotes.html (last viewed September, 2003).

Anne W. Branscomb. *Who Owns Information?* HarperCollins Publishers Inc., 1994.

Saacha Brostoff and M. Anegela Sasse. Are Passfaces more usable than passwords? A field trial investigation. *Proceedings of HCI 2000* (September 5-8, Sunderland, UK), 405-424. Springer, 2000.

L. Jean Camp, Cathleen McGrath, and Helen Nissenbaum. Trust: A Collision of Paradigms. *Proceedings of Financial Cryptography 2001*. Springer-Verlag, 2001.

L. Jean Camp, First Principles for Copyright for DRM Design. *IEEE Internet Computing*, 7(3):59-65, 2003.

L. Jean Camp. Design for Trust. *Trust, Reputation and Security: Theories and Practice*. Rino Falcone, ed. Springer-Verlag, 2003.

L. Jean Camp, Cathleen McGrath, and Helen Nissenbaum. Trust: A Collision of Paradigms. *Proceedings of Financial Cryptography 2001*, 91-105. Springer-Verlag, 2001.

L. Jean Camp and Rose Tsang. Universal service in a ubiquitous digital network. *Journal of Ethics and Information Technology*, 2(4):211-221, 2001.

L. Jean Camp & Ka-Ping Yee Human implications of technology, *Practical Handbook of Internet Computing* ed. M. P. Singh, CRC Press (New York, NY) Winter 2003.

Cheskin and Studio Archetype/Sapient. eCommerce Trust Study. January 1999.

David Clark and Marjory Blumenthal. Rethinking the design of the Internet: The end to end arguments vs. the brave new world. Telecommunications Policy Research Conference, Washington, DC, September 2000.

Computer Science and Telecommunications Board. *Trust in Cyberspace.* National Academy Press, 1999.

Lorrie Cranor, Designing a Privacy Preference Specification Interface: A Case Study. CHI 2003 Workshop on HCI and Security.

Lorrie Cranor and Joseph Reagle. Designing a Social Protocol: Lessons Learned from the Platform for Privacy Preferences. *Telephony, the Internet, and the Media.* Jeffrey K. MacKie-Mason and David Waterman, eds. Lawrence Erlbaum Associates, 1998.

Department of Defense. *Department of Defense Trusted Computer System Evaluation Criteria*. National Computer Security Center, 1985.

Rachna Dhamija and Adrian Perrig. Déjà Vu: A User Study Using Images for Authentication. *Proceedings of the 9th USENIX Security Symposium*, August 2000.

Jared Diamond. *Guns, Germs, and Steel: The Fates of Human Societies*, W. W. Norton & Company, 1999.

Whit Diffie and Susan Landau. *Privacy on the Line*. MIT Press, 1997.

Paul Dourish, Jessica Delgado de la Flor, and Melissa Joseph. Security as a Practical

Problem: Some Preliminary Observations of Everyday Mental Models. CHI 2003 Workshop on HCI and Security.

Elizabeth L. Eisenstein. *The Printing Press as an Agent of Change*. Cambridge University Press, 1979.

Carl Ellison, Improvements on Conventional PKI Wisdom. 1st Annual PKI Research Workshop, Dartmouth, New Hampshire, April 2002.

Carl Ellison and L. Jean Camp. Implications with Identity in PKI. http://www.ksg.harvard.edu/digitalcenter/conference/references.htm (last viewed September 2003).

European Union, Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995, *Official Journal of the European Communities*, L. 281: 31, 23 November 1995.

Federal Trade Commission, *Privacy Online: Fair Information Practices in the Electronic Marketplace*, Federal Trade Commission Report to Congress, 2000.

Charles Fischer. *America Calling: A Social History of the Telephone to 1940*. University of California Press, 1992.

BJ Fogg , Nicholas Fang, Jyoti Paul, Akshay Rangnekar, John Shon, Preeti Swani, and Marissa Treinen What Makes A Web Site Credible? A Report on a Large Quantitative Study. *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 61-68. ACM Press, 2001.

Batya Friedman, David Hurley, Daniel C. Howe, Edward Felten, and Helen Nissenbaum.

Users' conceptions of Web security: A comparative study. *Extended Abstracts of the ACM CHI 2002 Conference on Human Factors in Computing Systems*, 746-747. ACM Press, 2002.

Batya Friedman , Peter H. Kahn ,Jr ., and Daniel C. Howe. Trust online. *Communications of the ACM*, 43(12), 34-40, December 2000.

Batya Friedman, ed. *Human Values and the Design of Computer Technology.* CSLI Publications, 2001.

Batya Friedman and Lynette Millett. Reasoning About Computers as Moral Agents. *Human Values and the Design of Computer Technology*. B. Friedman, ed. CSLI Publications, 2001.

Simson Garfinkel, *Pretty Good Privacy.* O'Reilly, 1999.

Simson Garfinkel and Gene Spafford. *Practical UNIX and Internet Security, 2nd Edition*. O'Reilly, 1996.

Nathaniel Good and Aaron Krekelberg. Usability and Privacy: A study of Kazaa P2P file-sharing. *Proceedings of the ACM CHI 2003 Conference on Human Factors in Computing Systems*, 137-144. ACM Press, 2003.

Rebecca E. Grinter and Diane Smetters. Three Challenges for Embedding Security into Applications. CHI 2003 Workshop on HCI and Security.

Joseph R. Herkert (ed.) *Social, Ethical, and Policy Implications of Engineering : Selected Readings*, IEEE (1999)

Harry Hochheiser, Privacy, policy and pragmatics: an examination of P3P's Role in the

Discussion of Privacy Policy. Draft, 2003.

IEEE United States Activities Board. Position Statement on Encryption Policy. *The Electronic Privacy Papers,* 543. B. Schneier and D. Banisar, eds. John Wiley and Sons, New York, 1997.

Ian Jermyn, Alain Mayer, Fabian Monrose , Michael K. Reiter and Aviel D. Rubin. The design and analysis of graphical passwords. *Proceedings of the 8th USENIX Security Symposium*, August 1999.

Deborah Johnson. *Computer Ethics, 3rd ed*. Prentice Hall, 2001.

Mike Just. Designing Secure Yet Usable Credential Recovery Systems With Challenge Questions. *CHI 2003 Workshop on HCI and Security*.

Clare-Marie Karat. Iterative Usability Testing of a Security Application. *Proceedings of the Human Factors Society 33rd Annual Meeting,* 273 - 277, 1989.

Daniel V. Klein. Foiling the Cracker - A Survey of, and Improvements to, Password Security. *Proceedings of the second USENIX Workshop on Security*, 5-14, 1990.

David S. Landes. *The Wealth and Poverty of Nations: Why Some Are So Rich and Some So Poor*. W. W. Norton & Company, 1999.

Carl E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi. A Taxonomy of Computer Program Security Flaws, with Examples. *ACM Computing Surveys*, 26(3):211-254. September 1994.

Larry Lessig. *Code and Other Laws of Cyberspace*. Basic Books, 1999.

Larry Lessig. Tyranny in the Infrastructure. *Wired*, 5(7), 1997.

L. Jean Camp & Ka-Ping Yee Human implications of technology, *Practical Handbook of Internet Computing* ed. M. P. Singh, CRC Press (New York, NY) Winter 2003.

Walter Maner. Unique Ethical Problems in Information Technology. *Science and Engineering Ethics*, 2(2):137-154. February 1996.

Marshall McLuhan. *The Gutenberg Galaxy: The Making of Typographic Man*. Toronto: University of Toronto Press, 1962.

James H. Moor. What is Computer Ethics? *Metaphilosophy*, 16(4):266-275, October 1985.

William Mosteller and James Ballas. Usability Analysis of Messages from a Security System, *Proceedings of the Human Factors Society 33rd Annual Meeting*, 1989.

Nicholas Negroponte. *Being Digital - The Road Map for Survival on the Information Superhighway.* Alfred A. Knopf, Inc., 1995.

Jakob Nielsen. Usability engineering at a discount. *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, 394-401. G. Salvendy and M. J. Smith, eds. Elsevier Science Publishers, Amsterdam, 1989.

Andrew M. Odlyzko. Privacy, economics, and price discrimination on the Internet. *Proceedings of ICEC '03*. ACM Press, 2003 (forthcoming).

Andy Oram, *Open Sources: Voices from the Revolution*. O'Reilly & Associates, 1999.

Nathanael Evans, Avi Rubin, and Dan Wallach. Authentication for Remote Voting. CHI 2003 Workshop on HCI and Security.

John Pierce and Michael Noll. *Signals: The Science of Telecommunications*. Scientific American Press, 1990.

Ithiel De Sola Pool. *Technologies of Freedom*. Harvard University Press, 1983.

Neil Postman, *Technopoly: The Surrender of Culture toTtechnology*. Vintage Books, New York. 1996.

Nedra Prickler, Mechanics Struggle with Diagnostics. AP Wire, 24 June 2002.

Eric Raymond. *The Cathedral and the Bazaar*. O'Reilly, 1999.

Paul Resnick. A Response to "Is PICS the Devil?" *Wired*, 5(7), July 1997.

Jens Riegelsberger, M. Angela Sasse and John McCarthy. Shiny Happy People Building Trust? Photos on e-Commerce Websites and Consumer Trust. *Proceedings of the ACM CHI 2003 Conference on Human Factors in Computing Systems* (April 5-10, Ft. Lauderdale, Florida), 121-128. ACM Press, 2003.

Jerome H. Saltzer and Michael D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE,* 63 (9) 1278-1308, 1975.

M. Angela, Sasse, Sacha Brostoff , and Dirk Weirich. Transforming the weakest link - a human/computer interaction approach to usable and effective security. *BT Technology Journal*, 19(3), 122-131, July 2001.

M. Angela Sasse, Computer Security: Anatomy of a Usability Disaster, and a Plan for Recovery. CHI 2003 Workshop on HCI and Security.

Ben Shneiderman. Designing Trust into Online Experiences. *Communications of the ACM,* 43(12):57-59, December 2000.

Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612-613, July 1979.

Alexander Solzhenitsyn. The Law Becomes A Man. *Gulag Archipelago*. Little, Brown,

1975 (English translation).

Lee Sproull & Sara Kiesler, *Connections*, MIT Press; (1992).

Richard A. Spinello (ed.) *Case Studies in Information and Computer Ethics*, Prentice Hall; (1996)

Richard Stallman. The GNU Manifesto. http://www.fsf.org/gnu/manifesto.html, 1984 (last viewed September 2003).

Richard Stallman, Can You Trust Your Computer? http://www.gnu.org/philosophy/no-word-attachments.html, posted October 2002 (last viewed September 2003).

Serena Syme and L. Jean Camp. Open Land and UCITA Land. *ACM Computers and Society*, 32(3): 86-101.

George Trublow. *Privacy Law and Practice*. Times Mirror Books, 1991.

Robert C. Tucker, ed. Marx and Engels to Babel, Liebknecht, Branke, and Others. *Marx-Engels Reader,* 549-555. W. W. Norton, 1978.

Carl Turner, Merrill Zavod, and William Yurcik. Factors That Affect The Perception of Security and Privacy of E-Commerce Web Sites. *Proceedings of the 4th International Conference on Electronic Commerce Research* (November 2001, Dallas, Texas), 628-636.

John Viega, Tadayoshi Kohno, and Bruce Potter. Trust (and mistrust) in secure applications. *Communications of the ACM* 44(2):31-36, February 2001.

A. Whitten and J. Douglas Tygar. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. *Proceedings of 8th USENIX Security Symposium*, 1999.

L. Jean Camp & Ka-Ping Yee Human implications of technology, *Practical Handbook of Internet Computing* ed. M. P. Singh, CRC Press (New York, NY) Winter 2003.

Alam Whitten and J. Douglas Tygar. Safe Staging for Computer Security. *CHI 2003 Workshop on HCI and Security*. 2003.

John D. Woodward, Katherine W. Webb, Elaine M. Newton et al., Appendix A, "Biometrics: A Technical Primer," *Army Biometric Applications: Identifying and Addressing Sociocultural Concerns,* RAND/MR-1237-A, S RAND; 2001.

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

Dennis Wixon , Karen Holtzblatt and Stephen Knox. Contextual Design: An Emergent View of System Design. *Proceedings of the ACM CHI 1990 Conference on Human Factors in Computing Systems*, 329-336.

Ka-Ping Yee. User Interaction Design for Secure Systems. *Proceedings of the 4th International Conference on Information and Communications Security* (December 2002, Singapore).

Mary Ellen Zurko and Richard T. Simon. User-centered security. *Proceedings of the UCLA Conference on New Security Paradigms* (September 17-20, 1996, Lake Arrowhead, California), 27-33.