

# Transparency and Trust in Computational Systems

Rebecca Mercuri<sup>1</sup>  
Radcliffe Institute for Advanced Study  
Harvard University

## Abstract

*Transparency and trust are inherently intertwined concepts but their relationship is not well understood. Some believe that trust is enhanced by secrecy, while others maintain that obscurity reduces security. The example of anonymous voting is used to illustrate the manner in which external auditability can provide necessary assurances in computer-based transactions. Transparency issues related to standards, cryptography, intractability, usability, authentication, and risk are examined.*

## 1 Introduction

The slightly tarnished image of the computer industry, in terms of its ability to maintain security and privacy for business and private users, has played a role in impeding the growth and acceptance of e-commerce in recent years. Despite projections of banner e-sales for the 2004 holiday season, there has not yet been full recovery from the doldrums of the post-dot-com era. Similarly, e-government initiatives, such as electronic voting, have met with resistance despite the enthusiasm of vendors and officials. On-line confidence in products and services has eroded to the point where “use it at your own risk” is the general assumption in many cases.

One reason for this mistrust may be the perception (or actuality) that hosts and providers have lost control of the digital data transport medium as well as the software infrastructure that supports it. The barrage of spam emails (especially those with spoofed requests urging users to make password changes), along with the constant need to patch security holes in operating systems and applications programs, continues to have a corrosive effect on trustworthiness.

The fact remains that computers, particularly on the Internet, are increasingly subjected to a variety of aggressive attacks for which none but palliative or patchwork solutions have been presented. Consumers have grown increasingly skeptical of hype-tech or geek-speak security assurances. Lock icons, digital signatures, privacy policy statements, and other techniques used to provide security assurances are generally scoffed at or deemed an annoyance by most users. For example, pop-up security warnings are turned off, or left on for reasons that amount to not much more than just superstitious beliefs.

The result is that, on a daily basis, information technology departments are forced to walk the tightrope between trying to provide timesaving automation services and dealing with costly downtimes, with the winds of unpredictability at their backs. That sometimes even their best safety nets fail is depicted in TV ads displaying vacuous security promises by large corporations to beleaguered IT staff members. Yet none of this addresses the actual problem, which is that the customer has no obvious way of determining how to trust the systems they choose or are required to use.

## 2 What is Trust?

At the root of all of this, lie general questions about the nature of trust, a utopian ideal that is not well defined or even well understood. The problem is compounded when attempts are made to implement or integrate such vague notions into computational or rule-based systems.

The word trust itself has varying meanings, and many of these are invoked when we use the phrase “trusted computing.” The overriding definition is that of reliance or dependence, as in “I trust that you will ...” but there is also some optimism or hope in the future, as with “we can

---

<sup>1</sup>mercuri@acm.org  
<http://www.notablessoftware.com>

trust that the outcome should ...” Certainly the concept of custody or care is also involved, with “placed in the trust of ...” emphasized in beneficiary relationships (as when saying “the corpus of the trust goes to...”). And trust abounds in commercial or social entity relationships, such as charitable trusts, land trusts, bank trusts, and living trusts.

Choosing to trust often involves a mystical transcendence – the phrases “blind trust” and “absolute trust” come to mind. Indeed, all United States currency bears the words “In God We Trust” (provoking the old joke that it is implied that “all others must pay cash”). When situations are too complex for humans to comprehend, there is salvation in the idea of placing trust in a deity – as in Proverbs 3:5-6:

*“Trust in the Lord with all your heart  
and lean not on your own  
understanding; in all your ways  
acknowledge him, and he shall direct  
your paths.”*

This “all knowing, all powerful” transfer of trust is increasingly yielded to computers, even when evidence of reliability or safety is sorely lacking.

Individuals in whom we place trust include authorities, experts, officials, caretakers, and those we are familiar with or have proximity to (such as family or friends). We ask for references, assuming that trust can be passed along in a daisy-chain fashion. We build trust through brand loyalties, and we are willing to grant trust (newcomers start out with more than just zero) and provide forgiveness when trust is breached.

Many of these aspects of human nature can be exploited through cons like “Ponzi schemes” and other affinity fraud tactics – including the ways that ratings on Amazon and eBay have been manipulated to indicate unjustified trust levels. Our uses of the terms “artificial intelligence” and “expert systems” further enhances the sense of trust in computers that are typically no more knowledgeable, and often less so, than the fallible humans who designed them.

When trust is questionable, candor often plays a role in providing assurances. The idea that transparency or openness increases trust is embodied in the word “antitrust” – the creation of increased confidence through the process of breaking up business trusts that might privately engage in practices that discourage marketplace competition. In a similar fashion in governments,

competitive interests are served and trust is enhanced through policies that reveal hidden agendas, such as sunshine laws and the U.S. Freedom of Information Act.

### 3 Closed vs. Open Source

These opposing concepts of “trust me” versus “trust yourselves” along with transparency, or lack thereof, are certainly evident in the various camps of software developers.

At one extreme, we have the legacy view of Security by Obscurity [1], a philosophy that maintains that by concealing source code and design, one can prevent or minimize malicious activity. The ongoing proliferation of malware, such as the Blaster worm (the 32K-byte package that included the deposit of backdoor Trojan Horse software for later use) that, over a couple of days, infected more than a half-million Windows-based computers world-wide, is proof positive that the closed source technique is no more secure than the vast border between Canada and the USA.

The opposite side is characterized by the Open Source movement, where community review is viewed as a way of ferreting out and correcting software flaws that might otherwise have been exploited. Although Open Source supporters claim that their products appear to suffer fewer problems, this may be attributed more to their smaller percentage of the marketplace than to an inherently rugged nature, as indeed, the number of Linux worms and other open source attacks continue to grow.

### 4 Auditability

Yet, despite ongoing concerns about integrity and security, we trust large quantities of critical data and processes to computational systems. File types and media formats are continually in flux, often as the result of software security updates. The challenge of keeping information compatible and accessible is a nightmare for archivists and users alike – and an impetus to those who wish to crack their (often proprietary) codes. Considerable decision-making relies on collections of data that must be accurate and reliable, so additional confidence is typically provided through redundancy and auditability.

The double-entry model for accounting is illustrative of this methodology – separate sets of books are independently maintained and then cross-checked for accuracy by auditors who are deemed competent and trustworthy. Layers of assurance are therefore created by the systematic process used, with the expectation that flaws will be exposed and mitigated. Clearly, this does not always succeed, as evidenced by the rash of fraudulent accounting practices uncovered in the early part of this century.

The problem arises when internally conflicting goals (such as profit incentives) reduce the amount of transparency necessary in order to apply the auditing process appropriately. Since computers, like people, are inherently somewhat opaque, one must ascertain whether the level of transparency provided is sufficient to ensure trust in the system.

One might view this balance between transparency and trust in terms of a concentration or broadening of risk.[2] With Security by Obscurity, transparency is deemed inversely proportional to trust, and the risk is focused on a few (perhaps thousand, as with Microsoft™) employees. With Open Source, transparency is roughly equivalent to trust (or at least it provides a great deal of it), and risk is spread globally.

## 5 Intractability

As it turns out, neither closed nor open source examinations can provide total assurance of program correctness, because of the computational complexity issues that make it infeasible to determine that computer software will perform only the tasks it was designed for, and no more. Quite simply, it is impossible to differentiate the code you want running in a program from the code you don't want, on a generic basis. As Ken Thompson [3] admonished in his classic 1984 Turing Award lecture:

*"You can't trust code that you didn't totally create yourself. (Especially code from companies that employ people like me.) No amount of source-level verification or scrutiny will protect you from using untrusted code."*

His essay, titled "Reflections on Trusting Trust," goes on to explain how code could be embedded in the compiler (the program that prepares software to be run on a computer), or

the compiler that compiled that compiler, and so on. The nefarious code would thus be extremely difficult to find, but very easy to trigger, if one knows how it can secretly be accessed.

## 6 Certification

Somewhere in the middle, between open and closed source, we have certifications (such as the ISO Common Criteria [4] and the National Institute of Standards and Technology's Digital Signature and Secure Hash [5] standards) that provide imprimaturs used to ascribe confidence in the security of software products, the processes used to develop them, and their correct embodiment in distributed units.

This can be seen in the original purposes of the U.S. Department of Defense's Trusted Computer System Evaluation Criteria (TCSEC) program [6] (later superseded by the Common Criteria):

- "To provide a standard to manufacturers as to what security features to build into their new and planned, commercial products in order to provide widely available systems that satisfy trust requirements (with particular emphasis on preventing the disclosure of data) for sensitive applications."
- "To provide DoD components with a metric with which to evaluate the degree of trust that can be placed in computer systems for the secure processing of classified and other sensitive information."
- "To provide a basis for specifying security requirements in acquisition specifications."

Mandated by U.S. Congress with the Computer Security Act of 1987 (Public Law 100-235), TCSEC was originally applied to "sensitive information" whose "loss, misuse, or unauthorized access to or modification of which could adversely affect the national interest or the conduct of Federal programs, or the privacy to which individuals are entitled". The program and its successor (administered by NIST's Computer Security Resource Center) were also adopted voluntarily in other settings, such as health care and banking.

As with many government projects, though, the devil is in the details. In this case, the bottom-up approach used in TCSEC (that of creating security levels, somewhat analogous to security clearances for personnel, associated with features that require trust assurances) was too inflexible to accommodate the later modular and object-oriented designs. TCSEC's step-wise approach meant that the security bar would be globally set too high, or not high enough, for the entire system in an effort to accommodate worst-case scenarios without overly compromising realizability. As an improvement, the Common Criteria evolved as a top-down or "Chinese menu" through the process of identification of security features and their associated risks.

But this newer method, by allowing the vendor or purchaser to specify the details of a Protection Profile to be used for the Target of Evaluation, suffers from the problems of unintended and overlooked consequences. For example, although the Common Criteria deals adequately with numerous inter-dependencies (such as if you implement X, then you must implement Y and perhaps also Z), it fails to include any mappings for counterindications (if you implement J then you cannot also implement K or L).[7]

Ultimately, any metric for evaluation using a trusted computing approach is tied to the understanding, intuition, and honesty of the system designers and evaluators in providing an appropriate selection of assurance components. Unfortunately, with the Common Criteria, often the resulting Protection Profile is necessarily so complex and detailed that (other than perhaps as a checklist for certification or procurement) reliance on it for comprehensive security guarantees is unlikely, except for the most simplistic of systems. This is not to say that the program is useless, just that it has shortcomings, and many of those stem from the obfuscation of clarity through the imposition of a nest of details. Hence, it suffers from a lack of transparency, which is ironic, since the exercise of elaborating the details was obviously intended to provide such.

## 7 An Illustration: Anonymous Voting

Many of these issues are embodied in the ongoing debate (that I am pleased to have played a major role in starting) involving the availability

(or not) of voter-verified paper ballots with fully-electronic voting systems, in order to perform a trustworthy recount of election results.[8] The voting problem is a classic case of an inherent conflict of criteria, in that the requirement for anonymity means that access control and keypad audits are precluded from full implementation during the most critical data collection operation, that of ballot casting during the time of the election. It is this conflict that requires that external (manual) forms of auditing be used, via the polling book for access, and through the use of printed ballots for vote verification and totaling.

I assert that the lessons learned from the U.S. Presidential election in Florida 2000 should have been two-fold:

- 1) During the election, voters must be able to independently confirm that they have cast their ballots the way they intended to vote.
- 2) Following the election, there must be an indisputable way of determining the vote totals from the collected and secured set of cast ballots.

The major voting system manufacturers (and even some smaller newcomers) have maintained that electronic equipment can be trusted to collect and tabulate ballots properly. Such systems are increasingly being used to determine election outcomes. In 2004, some 80% of U.S. ballots were counted by computers, up from around 25% only a quarter-century ago. As well, nearly 30% of US votes, and 100% of India's, were collected this year on "black box" voting systems, where recounts were nothing more than reprints of information held inside of devices shrouded by proprietary non-disclosure agreements. But a growing body of scientists have endorsed the need for independently verifiable elections, and legislators, such as former Princeton University physicist and now U.S. Representative Rush Holt [9], have introduced state and federal bills that would mandate the availability of voter-verified paper audit trails.

Yet many fear the return to paper-based elections, not only due to their chad-filled past, but because of issues as to whether people can be trusted as well as (or better than) computers to

collect and count ballots. Elections are sociological phenomena for which technological solutions are being applied, whether they be paper and pencil, punchcard, or touchscreen. These technologies necessarily result in a disparity between the expectations for the voting system and what performance is actually capable of being delivered.[10] For example, election officials are quick to assert that “every vote counts” even though it has long been known that between 3-5% of votes may not be recorded at all in many elections, no matter what form of balloting technology was actually used.[11] Given such a high “missing” vote rate, it is easy to see why citizens lack confidence in the published election results.

Citizen fears and anxieties are inevitably compounded when the election process lacks transparency. The use of trade-secret protected equipment, procured and inspected in a non-public fashion, raises eyebrows. When such equipment fails and courts disallow independent testing, conspiracy theories begin to abound.

Those who are selling and purchasing these systems have done little to address security issues through recognized programs. To date, not a single voting equipment vendor has voluntarily complied with either TCSEC or the Common Criteria, not even to their lowest levels. Nor has any government or overseeing agency required them to do so. Attempts to place a Protection Profile document into the IEEE’s working draft for voting system standards have been stonewalled for years. And vendors continue to demand blanket protections from inspection for non-secure commercial-off-the-shelf components (such as wireless transceivers) even when these pose high risks in systems that provide no independent mechanisms to ensure the correctness of ballot contents and election results. On the positive side, the increased attention to this issue has stimulated research on some promising new voting system designs.

## 8 Cryptography

In voting, as well as other transactional applications, cryptographic approaches have been considered to enhance security. Indeed, David Chaum has claimed that it should be possible, using mathematical techniques, to provide total assurance of election results with methods that are independently verifiable.[12]

He has even devised an ingenious voting scheme, using overlays to display ballot choices, and mix-nets to maintain anonymity, but it should be noted that this method necessarily includes a way in which voters can confirm their selections on a tangible and human-readable medium. Berry Schoenmakers describes this general technique as used in election systems [13] as having two main stages:

- “First, voters post homomorphically-encrypted ballots, accompanied by a digital signature. Anyone is allowed to see the ballots cast, and anyone may verify the signatures for these ballots.”
- “Then, the product of all encrypted ballots is formed and only this product is decrypted, which yields the final tally. The product as well as its decryption (and hence the final tally) can be verified by anyone.”

But cryptography itself poses a host of transparency problems. First, there are the cryptographic keys that must be distributed securely, maintained in such fashion that disallows collusion among administrators (referred to, of course, as “trustees”) while preventing interception. Then there is the algorithm itself, which must be subjected to a thorough correctness proof, even if it appears to be verifiable (such as through the use of homomorphism). The issue of obsolescence of the algorithm, due to increased computational speeds and novel cracking approaches must continually be assessed. Finally, the implementation of the algorithm in software and/or hardware must also be demonstrated to be correct through rigorous, end-to-end provability. Under the Common Criteria program, satisfaction of these constraints would require certification at Evaluation Assurance Level 7, which no product has yet attained.

Furthermore, it is likely that none of this assessment process will be comprehensible by average (or even somewhat above average) end-users and administrators, so additional transparent assurances must be provided to indicate that the embodiment has not been altered from the approved version. It is therefore imperative to consider and mitigate the impact of this lack of transparency on the human trust of

the systems in which cryptographic processes are deployed. Most users are now savvy or skeptical enough to know that simple statements like “we use cryptography to secure your data” are no longer acceptable consolations, especially if problems arise.

## 9 Man-in-the-Middle

The transparency issues of cryptography aside, one of its other shortcomings is that although the parcel of information it is used to encode may be secure during data transmission, the ends (before and after encryption) may not be, and these are still vulnerable to attack. Conversely (whether cryptography is or is not present), users may be led to believe that the middle of a process is secure, as when they are unwittingly using a spoofed or compromised website to enter data. This problem with rampant theft of credit card information in online transactions led Visa to implement their Cardholder Information Security Program and MasterCard to create a Site Data Protection standard. Smaller outlets that do not have the capability to secure their own website can contract with third-party services that can act as filters between customers and credit agencies.

While e-sales growth was up 25% in 2004 with projections of another 35% in 2005, security events in 2004 increased by 150%. VeriSign now handles more than 250M daily via their firewall, intrusion detection, and intrusion prevention systems, and the Anti-Fraud Alliance estimates 1,000 new phishing attacks each month across the Internet. Some of the major security providers are now pooling information resources through this Alliance [13] in order to counter brand name hijacks. Although this middle-man approach may reassure creditors and corporations somewhat, it is only a stop-gap technique that does not address the overriding problems of online fraud and identity theft. Since anti-fraud heuristics are usually applied privately, it may not be possible to assess the amount of false positives and the impacts on fair access to services and user privacy.

If the trust needed for e-commerce solutions is increased by sacrificing trust in the customers, this inverse relationship will inevitably become too adversarial to survive. Customers who are subjected to absurd identity checks (such as reciting the three-digit number on the back of

credit cards, or mailing address change confirmation letters to one’s former location where a nefarious ex-roommate may still reside) like stooges in a security shell-game will use other providers that are viewed as less antagonistic. Other solutions must be sought.

## 10 Trust and Risk

Perhaps security folk have been addressing this problem from the wrong direction, by first assessing risk and then devising controls to mitigate it in order to earn trust. What we may instead need to do is assess trust, and then determine appropriate levels of risk. As J.P. Morgan observed, “...a man I do not trust could not get money from me on all the bonds in *Cristendom*.” The topsy-turvy view is that trust enables risk-taking. Economists and sociologists refer to this as “social capital.”[15]

This type of capital can be provided with threshold cryptography, where risk is managed by distributing trust to multiple parties. The Secure Electronic Transaction (SET) system used by MasterCard and Visa applies this method in order to force adversaries to take more risks by penetrating numerous systems rather than just one.[16] Of course layering and middle-man approaches also increase complexity which necessarily reduces transparency, so there may be diminishing returns to this technique.

## 11 Full Disclosure

That transparency might need to be managed in order to remain competitive is a concept that strikes a discord with legacy firms used to doing business the old-fashioned way, behind closed doors. But trust and transparency are not necessarily synonymous with full exposure, as Don Tapscott and David Ticoll explain in their book “The Naked Corporation.”[17]

Rather than appear as an Emperor without clothes protected by only the gossamer of public relations spin, some organizations are instead choosing to develop a “open kimono” culture by proactively engaging in transparency assessments and adjustments. As Tapscott said, “if corporations are going to be naked, they’d better be buff.” His comments stress that “undressing for success” cannot merely be a veneer, but rather it requires abiding by basic values in all operations – telling the truth,

honoring commitments, considering stakeholder interests, being candid about shortcomings, and building and delivering the best products. This is good advice for day-to-day operations as well as disclosures involving security matters.

## 12 Transparency and Usability

Certain dimensions of transparency are intertwined with usability issues. Andrew Patrick and Sabrina Mu [18] have identified some features as follows:

- *Understandability*: The application structure, navigation, procedures, features and terminology should be comprehensible for users.
- *Learnability*: The usage of the application or hardware device should exhibit a gradual learning curve and encourage exploration.
- *Self-descriptiveness*: When the application is presented to the user, it should be intuitively obvious how the system operates and what kind of tasks can be achieved.
- *Feedback*: This refers to whether there is an accessible, clear, and timely indication and response to user's actions.
- *Metaphors*: To support the transfer of real world knowledge into applications.

Although Patrick and Mu assessed these aspects in conjunction with the deployment of biometric security devices, they have general applicability to other products when improvements in transparency are being considered.

The United Nations Administration and Cost of Elections (ACE) project [19] has also identified various transparency issues related to technology implementations, including: accuracy, cost-effectiveness, efficiency, ethics, flexibility, inclusiveness, privacy, security, serviceability, and sustainability.

That improvements in these areas can be made is confirmed by three Australian studies that showed programmers equating trust with control at higher levels than non-technologist computer users.[20] They discovered a tendency by academic computer scientists to use the word

“control” almost as a synonym for “trust” – “control over the code, control over the likelihood of the technology breaking down, control over the interface, and over personal information and space.” For ordinary users, control pertained to their own and others’ access and regulated use, but trust was granted implicitly to the computational devices or processes.

Further investigation revealed that the scientists frequently designed for control rather than for ease of use, and that they occasionally mistrusted the technology so much that they did not use applications software that they had participated in designing. Indeed, familiarity may breed contempt, but this may become a counter-productive over-reaction if applied to technology. It also can undermine the confidence that is given to industry warnings – for example, many believe that Y2K problems never existed (since no calamity occurred) and were overblown to generate revenue for programmers, whereas it was the work done by the industry in preparation for the millennium change that actually averted many major problems.

One interesting view of trust involves the interactive effects of travail on its development. Helen Nissenbaum considers the type of trust that is developed from shared experiences, especially ones that are intense in nature.[21] Obvious examples are found in reality-TV shows, and corporate team-building events, where participants leave with a feeling of increased trust and bonding. Things that are too easily obtained are viewed as somehow less worthy, and there is often an additional intrinsic value, even if only a perceived one, in having overcome hurdles to attain a goal. To the extent that security hurdles sometimes impede ease of use, if not overly cumbersome, they may have the inadvertent side-effect of increasing travail, so perfect usability may not be a desired goal.

## 13 Conclusions

Clearly, transparency plays an increasingly important role in the world of computer security. But as with many sociological interactions with technology, the optimal balance is difficult to quantify. The consideration of a user-centric approach may help achieve the transparency needed to ensure confidence and reduce perceived risks in transactional experiences.

## References

- [1] Neumann, Peter G. and Mercuri, Rebecca T., "Security by Insecurity," Inside Risks, Communications of the Association for Computing Machinery, Vol. 46, No. 11, November 2003.
- [2] Landis, Lynn, private correspondence, September 2003.
- [3] Thompson, Ken, "Reflections on Trusting Trust," Communications of the Association for Computing Machinery, Vol. 27, No. 8, August 1984.
- [4] International Standards Organization, Common Criteria for IT Security Evaluation, August 1999. <http://csrc.nist.gov/cc/>
- [5] National Institute of Standards and Technology, Secure Hash Standard, April 1995. <http://www.itl.nist.gov/fipspubs/fip180-1.htm>
- [6] U.S. Department of Defense, Trusted Computer System Evaluation Criteria, December 1985. <http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.pdf>
- [7] Mercuri, Rebecca, Uncommon Criteria, Communications of the ACM, Vol. 45, No. 1, January 2002.
- [8] Mercuri, Rebecca, "A Better Ballot Box?," IEEE Spectrum, Vol. 93, No. 10, October 2002.
- [9] Holt, Rush D., "Introduction of the Voter Confidence and Increased Accessibility Act of 2003," U.S. Congressional Record, Extensions of Remarks, May 23, 2003, E1081-2.
- [10] Neumann, Peter G., and Parker, Donn B., "A Summary of Computer Misuse Techniques," 12<sup>th</sup> National Computer Security Conference, October 1989.
- [11] Caltech/MIT voting Technology Project, "Voting: What Is, What Could Be," July 2001.
- [12] Chaum, David, "Secret Ballot Receipts and Transparent Integrity," May 2002.
- [13] Schoenmakers, Berry, "Compensating for a lack of Transparency," Proceedings of the 10<sup>th</sup> Conference on Computers, Freedom and Privacy, April 2000.
- [14] The Anti-Fraud Alliance, <http://www.antifraudalliance.com>
- [15] Farrell, Chris, "Sound Money," June 8, 2002. <http://soundmoney.publicradio.org>
- [16] Frankel, Yair, and Yung, Moti, "Risk Management using Threshold RSA Cryptosystems," Usenix, 1998.
- [17] Stepanek, Marcia, "Don Tapscott on Transparency," CIO Insight, October 2003.
- [18] Patrick, Andrew, and Mu, Sabrina, "Usability and Acceptability of Biometric Security Devices," National Research Council of Canada, September 2004.
- [19] United Nations, Administration and Cost of Elections Project. <http://www.aceproject.org>
- [20] Singh, Supriya, and Christine Satchell, "Trust, Control and Design - A Study of Computer Scientists," RMIT University, Melbourne, Australia, August 2003.
- [21] Nissenbaum, Helen, "Securing Trust Online: Wisdom or Oxymoron?," Computer Ethics: Philosophical Enquiry, New York University School of Law, 2000.