ELECTRONIC VOTE TABULATION

CHECKS & BALANCES

Rebecca Mercuri

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania in Partial

Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2001

Norman I Badler

Dr. Norman I. Badler
Supervisor of Dissertation

Dr. Val Tannen for Val Tannen
Graduate Group Chairperson

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

IN MEMORIAM

Mae Churchill

Mentor, Friend, Defender of Democracy

First woman to receive the Ph.D. in Economics
from the Wharton School of the University of Pennsylvania, 1944.

I regret you were not here to enjoy the continuation of your efforts
and I offer this work with sincere thanks for your counsel and inspiration.

ABSTRACT

ELECTRONIC VOTE TABULATION

CHECKS & BALANCES

Rebecca Mercuri

Dr. Norman I. Badler

The subject of electronic vote tabulation involves a unique combination of technological, computational, and sociological problems that produce a set of constraints upon the systems used for ballot entry and vote counting. This document identifies the various types of voting systems; the hierarchy of constraints under which they are required to operate; and the numerous checks and balances that need to be provided in order to ensure accuracy and integrity. The thesis work involved a detailed assessment of the limitations of electronic vote tabulation systems using the framework of the ISO's Common Criteria. A minimal voting system was described, along with a procedure by which existing and proposed voting systems may be evaluated for potential flaws.

The result demonstrated the existence of a category of systems for which the Common Criteria can be deemed inadequate. The Criteria provides for assessment of system dependencies, but does not account for counterindications. Specifically, the requirement for ballot privacy creates an unresolvable conflict with the use of audit trails in providing security assurance. This has broad implications within other commercial arenas, particularly those involving anonymous data delivery. Other results involved an appraisal of possible election risks (such as global denial of service and Trojan horse attacks) that are enhanced by the deployment of electronic balloting systems, along with recommendations of considerations that can assist in reducing these vulnerabilities. A discussion of some issues related to the 2000 Florida Presidential election, recount, and litigation is included.

Table of Contents

List of Tables

List of Illustrations

Preface

In June 1989, as a committeewoman in Pennsylvania, I attended a meeting where Commissioner Lucille Trench announced that Bucks County was looking into the replacement of its lever-style machines with new electronic voting systems. As a computer scientist I felt uneasy about the prospect of turning our elections over to computer software and hardware, so when I returned home, I mentioned this to my then-husband, Patrick Mercuri, who recalled having read an article on this subject in The New Yorker magazine within the past year. We tracked the issue down, I photocopied Ronnie Dugger's lengthy essay and sent it, along with a three-page brief, to Commissioner Trench. In the brief I said the following:

> "As you know, our American system of government is based upon the principle of 'checks and balances.' The two-party system, the three branches of government, the division of Congress into the House and the Senate -- these were devised to guarantee that fair consideration be given to all issues that affect the populace. Computerized voting systems violate this essential principle because they are not, at present, adequately examined or controlled."

The Commissioner responded by requesting that I join their investigation and comment on the Shoup systems under procurement consideration. Thus began the journey that eventually resulted in this thesis. Along the way, I was able to convince Bucks County to retain its lever machines, and I also played a key role in ensuring that New York City did likewise with theirs. Although now more than a decade has passed, the words I wrote then still ring true, which is why I incorporated the phrase 'checks and balances' into the title of this work.

My quest to explain the inherent risks of computerization of the voting process to public officials has provided me with the opportunity to meet and interact with many experts in the field. I contacted the Computer Professionals for Social Responsibility's then-chair, Marc Rotenberg, who led me to Mae Churchill, the founder of Election Watch. She and her colleagues had investigated various computerized vote tabulation anomalies, and had provided commentary to the Federal Election Commission on their proposed voting equipment guidelines. Mae assisted my efforts in Bucks County, and I soon found myself included in her activities with the public hearings on voting systems in New York City. Through Mae, and in this work, I encountered a cast of characters, which often seemed to range from the ridiculous to the sublime, many of whom I greatly appreciate for their positive influence on my thinking through numerous discussions on this subject. These included: Ronnie Dugger who clued me to the occasional merits of writing with a pen (rather than a word processor) in the creative process; the late mathematician Irwin Mann, who argued with me that there must be a cryptographic solution for voting systems (despite his lack of success in conjuring up one); Roy Saltman who wrote about this subject long before it was fashionable; and Richard Wagner for providing insight on numerous aspects of election administration.

Among the many others who have supported my doctoral efforts along the way are: my mother, Zelda Provenzano for her stellar example of persistence (and success) despite occasional adversity; my many University of Pennsylvania Graphics Lab cohorts, especially Ranjit Bhatnagar, Pei-Hwa Ho, Jean Griffin, and Bond-Jay Ting, for their colleagueship, camaraderie and commiseration; and former RCA Labs researchers (Henry Baird, Doug Dixon, Deborah Hill, Mike Keith, Chris Kocher, and Dave Ripley) and too-numerous-to-mention current ACM/IEEE-CS Princeton Chapter fellows for their nagging and cheering (as appropriate). Penn State Professors Tom Warms and Frank Melone helped ensure the completion of my first Computer Science degree. They, along with Drexel Professors Hermann Gollwitzer and William Dowling, and my Great Uncle Ben and Great Aunt Shirley Serota all often encouraged me to finish this last one. My neighbor, Michelle VanNoy, is appreciated for her persistence in ringing my doorbell many mornings on her way to work to help ensure that I had an early start each day, during the final months of draft writing. Thanks also go to my new colleagues at Bryn Mawr College, especially Computer Science Department Chair Deepak Kumar, who generously offered an opportunity to practice my dissertation defense there. My apologies and thanks to anyone else I have neglected to mention here.

My thesis committee has played a large role in assisting me over numerous dissertation hurdles. I wish to thank my University of Pennsylvania advisor, Norman Badler, for his infinite patience with directing me through not one, but two theses (and for providing some funding) during my tenure at the School of Engineering and Applied Science.

Department Chair Mitch Marcus was instrumental in suggesting "why not write a thesis on your work with computerized voting?" when my directional hearing project stalled. Lyle Ungar listened to my complaining over the years about the Ph.D. process, and returned the favor by providing salient complaints of his own, on the early draft of this document. As well, Dave Farber's ongoing commentary on security and privacy issues has been equally helpful. I have reserved the last, but certainly far from least, acknowledgement for Peter Neumann, who served as outside advisor for this thesis. With Peter, over the last decade, I have provided testimonies, written articles, and served on panel sessions on the subject of electronic voting systems. His insight and guidance has been invaluable, from his painstaking notations regarding that which required excision from or inclusion into the text, to help with pointers to key resources and individuals when necessary, and the occasional pep talk when spirits lagged. I heartily appreciate the contributions of my committee members to this work and look forward to continued fruitful interactions with all of you from the other side of the Ph.D. fence.

Finally, in the tradition of Mae (Huettig) Churchill's closing paragraph of her dissertation, let me also state that "it remains only to add that none of the persons named is responsible for the various sins of commission and omission probably scattered throughout the text."

REBECCA T. MERCURI

APRIL 2001

Chapter 1

Democracy is only an experiment in government, and it has the obvious disadvantage of merely counting votes instead of weighing them.

-- Dean Inge

1.0 Thesis Structure

This thesis:

a) identifies the various types of voting systems currently in use;

b) addresses certain major unresolvable issues in electronic vote tabulation from the standpoint of theoretical computer science;

c) discusses the realizability of secure and accurate voting systems within a hierarchy of constraints;

d) provides a procedure by which existing and proposed voting systems may be evaluated for potential flaws;

e) demonstrates the existence of a category of systems for which the ISO's Common Criteria can be deemed inadequate for security assurance.

The format of this document is as follows:

Chapter 1: Statement of the problem and background information. Vulnerabilities and constraints. Various case illustrations.

Chapter 2: Technology. Description and comparison of common balloting systems -- lever machines, paper, direct recording electronic, vote-by-phone, Internet.

Chapter 3: Sociotechnology. Legislation pertinent to voting systems. Components relevant to voting systems and their inherent flaws: verification and validation, audit trails, open source software, encryption, access controls, trust, data issues, secure channels.

Chapter 4: Sociology. The election business. Misuse of balloting systems.

Chapter 5: Common Criteria. CC evaluation process, voting system criteria, evaluation levels and constraint conflicts.

Chapter 6: A minimal voting system. Tabulation units, CC evaluation, realizability under constraints. California Internet Voting Task Force evaluation.

Chapter 7: Conclusions. Recommendations, remediation, prognosis, future directions.

Chapter 8: Postscript. Lessons learned from Election 2000.

Essentially, this thesis is a claim in the form of "process X is a terrible way to do Y" which is a defendable dissertation statement as long as (a) the claim is new and (b) process X is a serious contender. Both (a) and (b) will be established in this document, with process X identified as the use of computational systems for voting and Y being electronic ballot tabulation under the six constraints (commandments) delineated by

Shamos (see Section 1.4). To paraphrase [GTECH]: "The defense will be an analysis of the limits of process X, i.e., things it can't do, or things it does wrong, along with evidence that those things matter."

This thesis work is distinguished from others, in that it provides a secure-system view that includes the treatment of anonymity. A major side result of the thesis is that it produces a test of the applicability of the Common Criteria process, which is important as there are very few real evaluations of it to date. As to the evidence that such things matter, one need only look to recent legislation, policies, and events (such as the latest e-commerce and Internet hacks) to see the growing need for impenetrable, reliable systems. Electronic voting had begun to receive attention during 2000 as some U.S. municipalities implemented online voting in primary elections, but it is now the subject of intense focus as many communities contemplate the replacement of their existing election systems in the aftermath of the Florida situation. As well, the United Kingdom and Australian governments are currently considering web-based voting for their general elections. A significant non-voting application for this dissertation involves the newly imposed requirements for the release of NSF-funded research data, where anonymous reporting would also be essential.

This thesis should be evaluated in terms of the comprehensive nature of the discussion. Have all significant avenues of vulnerability been adequately described? Has the balance between the sets of imposed constraints and assurable system levels been defined? Does the authentication process provide a methodology by which voting systems may be evaluated? Satisfying these questions yields an effective thesis, and also a document that can be used to improve the quality of electronic vote tabulation systems in the years to come. The security analysis presented can set an example for how to approach other integrity-sensitive applications as well.

It should be noted that this thesis was publicly defended at the University of Pennsylvania on October 27, 2000. I have chosen to maintain the form of this dissertation as it was then presented, since it now has merit both as a scientific document and also as an historic work which pre-dated the extensive legal, media, and academic investigations of voting systems which followed the November 7, 2000 election. Material specific to that election appears at the end of this thesis, after the conclusions and recommendations chapter. The thesis stands on its own without these additional writings, but the supplementary information should assist readers who may be contemplating the construction or procurement of new voting equipment, or who would like to know more about technical aspects pertinent to the November Presidential election.

1.1 Author's Contributions

For those readers unfamiliar with the subject of electronic vote tabulation, it may be helpful to describe here the prior contributions of this author (Doctoral Candidate Rebecca Mercuri) to the body of existing literature. For a considerable portion of the last decade, I have provided expert testimony on this subject in the form of commentary at public hearings, sworn statements, private meetings with equipment vendors and elected officials, written system evaluation documents, professional journal articles, and media interviews. (Most of these commentaries are available on my Website at: www.notablesoftware.com via the electronic vote tabulation link.)

In constructing this thesis, my own writing is used extensively. In particular, the description of the Sequoia Pacific system in Section 1.5, appeared earlier as [MER92a]. The discussion of the St. Petersburg, Florida and Ohio elections in that same section, was taken from [MER93b]. Sections 2.3 on DREs, 3.2 on verification and validation, and 3.3 on audit trails are largely from [MER92]. Section 4.1 on the election business is from [MER93] and Section 4.2 on misuse is from [MER93a]. For the Urban Policy Research Institute's Election Watch group, I produced a voting machine evaluation document, which provided the groundwork for some of the aforementioned pieces.[MER91]

Additional work included chairing the National Institute of Standards and Technology's 16th National Computer Security Conference session on Security and Auditability of Electronic Vote Tabulation Systems, where I played a key role in generating papers with the panelists. One of these was Peter Neumann's "Security Criteria for Electronic Voting," which eventually was reprinted in his Computer Related Risks book [NEU95] and which provides the framework for Section 5.2 on voting system criteria. Also, as chair for the session entitled Electronic Voting -- Threats to Democracy, at the 3rd Conference on Computers, Freedom and Privacy, I was again instrumental in the development of panelist papers, including those by Michael Shamos [SHA93] and Roy Saltman [SAL88], referenced in the body of this dissertation.

1.2 Background

Suffrage, the right to vote, has long been viewed as integral to the maintenance of elective democracies, such as that in the United States of America. Election systems are also important to many other foreign countries, stock-based companies, worker's unions, homeowner and community associations, and so on. Through the election process, eligible individuals are able to register their opinions (via referendum questions and preferential balloting) and they can elect representatives who are viewed as capable of reflecting the views of the majority while serving their constituents. Given the importance of the voting process, one might then surmise that the highest security methods would be required to be applied to any computer hardware and software used in elections, but this is presently not the case. Vote tallying equipment used for general elections in the U.S. is exempt from the Congressional Computer Security Act of 1987, despite the fact that it processes "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." [NCS91][SAL93]

Steadily and silently, computer hardware and software have become entwined with virtually every phase of the U.S. election system. Voter registration databases are now automated in nearly all municipalities, ballots are cast (either directly, or via punch-card and mark-sense ballots) into computer-based equipment in ever-increasing numbers, and end-of-day results are often transmitted electronically between municipal headquarters and to the media. Current and pending legislation will further enhance computer involvement with elections via motor-voter registration, vote-by-phone, Internet polling, and electronic ballot transmission from overseas military and civilian personnel.

As this technology evolves, many open questions emerge. What degree of accuracy should be demanded in voting systems? How much is the public willing to pay for increased security? Do auditability, confidentiality, assurance, system integrity, and process integrity issues impose a myriad of conflicting requirements? Is it possible to "throw" a national or local election via internal or external manipulation of hardware, software, and/or data? These and other related issues are addressed within the context of this thesis.

Although the voting topic may at first seem narrow in scope, there are broader computer applications that also require the accurate logging of a private transaction with a human. The Automated Teller Machine (ATM) comes immediately to mind, but other access-type situations (such as passkey entry into a secure facility, or permission to use a restricted database) are certainly viable. Additional activities that could conceivably have requirements similar to those handled by the voting process include: confidential calls to suicide or abuse hotlines, AIDS or other infectious-disease blood test reporting, Swiss-style banking transactions, and tracking of activities of top-secret personnel without revealing their identities. A recent application involves the Shelby amendment to the Freedom of Information Act, where the required release of all research data produced with federal support poses serious problems in studies involving promised confidentiality to participants.[BEA99] The suggestions and solutions provided herein could be extended to these and other similar types of systems.

One implementation of an electronic vote tabulation system is the Direct Recording Electronic (DRE) voting machine, which is used as an example in this thesis discussion. The DRE differs from the aforementioned applications due to the fact that, in addition to the requirements for accuracy and privacy, there is the mandated necessity to provide complete anonymity. In other words, banking and access applications can (in fact must) allow tracking back to the user of the system, but the DRE must ensure that such tracking is impossible. This creates an enigma in the verification process, which will be described in Section 3.3.

As it turns out, voting has other dissimilarities from automated banking. In a banking transaction, one is typically given a receipt that serves as an arbiter in case of discrepancy, say for a deposit that was not properly recorded. In the case of a withdrawal where the incorrect amount of cash is dispensed, other checking mechanisms (like the video monitor, and the audit of total cash in the machine) also are available. Plus, if a particular bank became well known for short-changing its customers at the ATM, it would likely lose business. With a voting system, the citizens have no choice as to which machines they may use, following the final equipment purchase and deployment in a municipality. One can select a different bank, but one can not simply go down the street to vote at a different polling place. Furthermore, banks are insured against losses, elections are not.

1.3 Vulnerabilities

Inherent in the nature of electronic vote tabulation (indeed, in all computers) are "gaps" that can be intentionally or accidentally used to subvert the systems. As identified by Peter Neumann [NEU89], these fall into three categories as follows:

The technological gap is that disparity between the expectations for the hardware and software, and what performance is capable of being delivered. In the case of voting systems, we might demand 100% accuracy from the ballot count, but actually, all existing electronic vote tabulating methods have unavoidable margins of error. This gap also applies to privacy of the balloting system (such as radio frequency emissions), resistance to tampering, as well as auditability, configurability, and operability.

The sociotechnical gap involves the differences between social policies and computer policies. Social policies generally take the form of laws (regarding computer crime, privacy, etc.) and codes of ethics and commerce. Such laws have lagged behind the rapid advances in technology, and may not adequately address many new issues that could arise. Say, for example, a voting machine is configured so that votes for candidate A are registered to candidate B. This is not fraudulent when performed in a laboratory demonstration or even in a test procedure. It is fraudulent, though, if intentionally done in an actual election, but there may exist no laws pertaining to such misuse. Furthermore, the determination of intent and fraud in the computer setting is difficult, if not (in some cases) impossible, to differentiate from simple errors or omissions.

The social gap is that between social policies and human behavior. It involves the possibility of misuse during the election process. Should the manufacturer be required to foresee all potential problems, and provide traps or flags to preclude these from happening, or should some of the responsibility for procedural correctness rest with the operators? How will the operators be selected and trained? Will the manufacturer be permitted access to the system during the election, and if not, what actions should be taken in the case of malfunction, and how shall repairs be provided while still assuring integrity? Currently no legislation prohibits foreigners or persons with criminal records from working on or manufacturing voting machines; who should have access?

These gaps provide a mechanism for addressing the vulnerabilities of electronic vote tabulation as follows: issues related to the technological gap are addressed in Chapter 2, sociotechnical matters are detailed in Chapter 3, and sociology is the focus of Chapter 4.

1.4 Constraints

Michael Shamos [SHA93], a voting machine examiner, has suggested a set of fundamental requirements for electronic voting systems, which are provided in the form of commandments listed in decreasing order of importance:

I. Thou shalt keep each voter's choices an inviolable secret.

II. Thou shalt allow each eligible voter to vote only once, and only for those offices for which the voter is authorized to cast a vote.

III. Thou shalt not permit tampering with thy voting
system, nor the exchange of gold for votes.

IV. Thou shalt report all votes accurately.

V. Thy voting system shall remain operable throughout each
election.

VI. Thou shalt keep an audit trail to detect sins against
Commandments II-IV, but thy audit trail shall not
violate Commandment I.

Shamos goes on to make some interesting observations about the Commandment list, such as the fact that accurate reporting is not the most important rule. In his experience, he has observed little or no tolerance of violations of Commandments I, II or III, yet violations of IV are even permitted in certain jurisdictions, particularly where the outcome of a race is unaffected by minor vote tally discrepancies. Ballot secrecy can be waived by the voter, such as in the case of absentee ballots. Some municipalities allow media to be present even within the voting booth area, and in such places a voter can elect to have his or her ballot casting filmed. It is common for parents to bring small children into the voting booth with them. Other waivers would include voter assistance for the physically challenged. For the most part, though, the requirement for privacy far supersedes the demand that every single vote be recorded and reported accurately. Commandment II has recently been made even tougher through the federal motor-voter reporting regulations, which track citizens' current residences through driver's license and vehicle registrations. Commandments V and VI are even less stringently enforced than the others above them. All of these Commandments are referred to subsequently in the framework of discussion for this thesis.

1.5 Illustrative Cases

It is helpful to review a few important cases involving electronic vote tabulation in order to provide some perspective on the issues and processes involved. Throughout the country, at virtually every election season, vote tabulation processes are questioned. A few that are notable for their level of egregiousness are described below. (It should be noted that this author served as an independent expert witness on both the New York City procurement and the St. Petersburg, Florida city election investigation described later in this section.)

On July 23, 1992, New York City Mayor Dinkins announced that 7,000 DRE voting machines would be purchased from the Sequoia Pacific company, pending the outcome of public hearings. This announcement contradicted the recommendation of the New York City Bar Association, independent groups of concerned scientists and citizens (such as Election Watch, Computer Professionals for Social Responsibility, and the New York Public Interest Research Group), and SRI International (a consultant to New York City, and the system evaluator). SRI's pre-procurement evaluation [BAE91] indicated that the systems failed 15 environmental/engineering requirements and 13 functional requirements, including resistance to dropping, temperature, humidity and vibration. Under the heading of reliability, the vendor's reply to the testing status report stated: "SP doesn't know how to show that the Electronic Voting Machine and its Programmable Memory Device meets [this] requirement -- this depends on poll workers' competence."

When a similar Sequoia Pacific system was examined prior to purchase in Pennsylvania, it was rejected for a number of reasons, including the fact that it "can be placed inadvertently in a mode in which the voter is unable to vote for certain candidates" and it "reports straight-party votes in a bizarre and inconsistent manner." When this report was brought to the attention of the New York City Board of Elections, they replied by stating that "the vendor has admitted to us that release 2.04 of their software used in the Pennsylvania certification process had just been modified and that it was a mistake to have used it even in a certification demonstration." Other problems noted with the system included its lack of a guaranteed audit trail and the presence of a real-time clock, which Pennsylvania examiner Shamos referred to as "a feature that is of potential use to software intruders."

Over the following years, numerous public hearings and trials of the Sequoia DREs indicated that the equipment was not yet fit for use. In an examination of the Security and Control document provided by Sequoia Pacific, SRI noted that the report was "inadequate and does not meet the requirements of the NYC/Sequoia Pacific contract."[NEU94] In particular, the document did not elaborate on the manner in which security controls would be provided, assessed, and ensured. Ultimately, the final purchase approval was withheld, and various lawsuits between the City and the system vendors ensued. Finally, during the Summer of 2000, an out-of-court settlement was reached, with the City agreeing to pay for only the single voting machine and services it had received, but allowing the remainder of the $60M contract to be cancelled.[MOO00] The result was that New York City had incurred unrecoverable costs of about $17M for the lengthy procurement process, without being able to replace its mechanical lever machines.[DUG00] Yet, despite the serious concerns raised by this evaluation, other United States municipalities have adopted or are considering DRE equipment for election use, some from the same vendor, so this matter is far from over. The New York situation was unique, in that extensive scrutiny of the systems occurred prior to adoption (based on highly detailed procurement specifications), rather than after election use, as in the next examples.

During the March 23, 1993 city election in St. Petersburg, Florida, two systems for ballot tabulation were being used on a trial basis. When checking the results after the election, it was revealed that in an industrial precinct in which there were no (zero) registered voters, the vote summary indicated 1,429 votes for the incumbent mayor (who incidentally won the election by 1,425 votes). Officials explained under oath that this precinct was used to merge regions counted by the two computer systems, but were unable to identify precisely how the 1,429 vote total was produced. Investigation by the Pinellas Circuit Court revealed sufficient procedural anomalies to authorize a costly manual recount, which eventually certified the results. The Florida Business Council, an independent group of concerned citizens, continued to look into this matter but was ultimately overwhelmed by lack of funding.

Equipment-related problems are a source of concern on election day, especially when time-critical operations must be performed. Ohio's Columbus Dispatch reported [STE92] that 40 of the 758 electronic machines used in Franklin County's June primary required service on election day. Noted is the fact that only 13 of the County's 1,500 older mechanical lever machines needed repair during the election. Defects reported for the electronic machines included: voter ballot cartridges not loading properly into the tallying computers, so those precincts' results had to be hand-keypunched; power boards on some machines containing blown fuses; and malfunctions with the paper tape on which the results were printed. Difficulties with the central software for merging the electronic and mechanical tallies created further delays in reporting results. Officials decided to withhold the final payment of $1.7M of their $3.82M contract until greater reliability was assured.

If Franklin County did not have enough trouble due to the above matter, two electronic ballot tabulation vendors contested the contract award. MicroVote Corporation sued the R. F. Shoup Corp., Franklin County, and others in the U.S. District Court for the Southern District of Ohio, Columbus Division, for over $10M in damages, claiming conspiracy and fraud in the bidding process.

In yet another region of Ohio, in the same primary, the Cleveland Plain Dealer [SAM92] reported that Kenneth J. Fisher, a member of the Cuyahoga County Board of Elections, allowed an employee to feed a computer a precinct identification card that was not accompanied by that precinct's ballots, during the vote tabulation process. Apparently, the ballots cast in the Glenville region had been inadvertently misplaced, and at 1 A.M. the board members "were tired and wanted to go home" so the election official authorized the bogus procedure, despite the fact that doing so might have constituted a violation of state law. Subsequent inquiry did not lead to any indictments.

Human errors can occur with computerized tabulation, as happened in the November 1998 Congressional election in the 12th District of New Jersey. This author happened to witness some of what transpired, when at the post-election party in Princeton, the vote tallies each of the poll workers had brought with them didn't jibe with the numbers being reported out of the Mercer County Clerk's office. Rush Holt refused to concede the election to incumbent Mike Pappas, and eventually County Clerk Catherine DiCostanzo reported the correct vote count, but only after Holt missed his opportunity to claim victory on the 11 P.M. news reports. The County system requires that the results faxed from each municipal district be hand-typed into a computer that totals the regions. DiCostanzo said that "the error happened once the numbers arrived at the Clerk's office and was due to inaccurate keystroking" and that "it was simple human error, there is a lot of numbers coming into this office and we had a key-in error." A last-minute recount named Holt the winner by an 8,827 vote margin (Holt received 51% of the total votes cast).[SHE98]

Due to the recent rapid deployment of Internet voting systems, various municipal and private elections have experienced difficulties. The Association for Computing Machinery (ACM) was one of those with egg on its face during its 2000 officer election. Postcards were mailed to members stating that: "Election Technology Corporation (ETC) has been contracted to handle the ACM election. We experienced some difficulties in the initial days of the electronic balloting - April 12th through April 17th. Consequently, some votes were not captured. Only 72 ballots had been cast during that period. Therefore, those ACM members who voted electronically during this timeframe are being asked to recast their votes." The card went on to say that ETC apologized for any inconvenience and assumed the cost of the mailing. One is left wondering if any other problems occurred with the election that perhaps were not detected and rectified.

Even more disconcerting were the problems, which occurred with the 2000 Arizona Democratic Primary election, the first in the country to use an Internet balloting format. This event was widely touted as a success because voter participation far exceeded expectations, even though actually only 10% of the State's registered Democrats voted (less than half of those via the Internet). (Incidentally, taking voting data out of context in order to make statements about elections that 'prove' a particular point, is a well-known tactic used by both media and candidates.) The Voting Integrity Project (VIP), a national, nonprofit, non-partisan voter rights organization, filed a lawsuit in an attempt to halt the on-line primary, but its request was denied. Later, they reported that "the election was completely run by election.com under contract to the Arizona Democratic Party and the system used was not certified or supervised by election officials." VIP has complained about the vendor's non-disclosure regarding integrity protocols, indicating that there was no protection from denial-of-service, virus, or Trojan horse attacks, although none were observed (this does not mean that none occurred). VIP also noted that: "Voter authentication was minimal and could, in some cases, have been easily defeated, leading to fraud. The election was completely vulnerable to insider violation of voter privacy -- election.com issued the PINs and had access to the ballots. Many Macintosh computers, and all computers using older Netscape browsers, were unsupported. There was an as-yet-unexplained one-hour total outage on the first day of the election."[PHI00] The website was not accessible by the visually impaired, a segment of the population likely to choose this method of balloting, thus in violation of the Americans with Disabilities Act. Weaknesses in voter authentication leaves the door open to fraud on the part of a vendor in order to possibly exhibit increased turnout and acceptance for their system -- although this is not suspected here, it is certainly an issue that must be addressed.

A fairly critical article in The Industry Standard, an Internet trade publication, noted the problems experienced with older browsers in the Arizona primary election and also that many Democrats did not receive their PINs and could not access the system.[LED00]

Other election concerns were described, such as disenfranchisement of minorities, and persons who voted more than once by intercepting the mailed PIN notification of others. Despite all of this evidence, election.com presents the appearance of being oblivious to these problems, to the point of even distributing the negative Industry Standard's article along with their promotional materials at the Political Fest (a trade show held in Philadelphia during the Republican Convention). In an interview, company CEO Joe Mohen was quoted as saying: "I think the fact that the security was flawless is really a testament to both our previous track record and I think it is giving people ... a level of comfort." In a bit of braggadocio, he also said: "We're giving the people the opportunity to change the world. There's a real sense that we are doing something sacred. We are giving people the opportunity to vote."[WSJ00] A few months later, election.com conducted another Internet election, this for the Board of Directors of the Internet Corporation for Assigned Names and Numbers (ICANN), the powerful policy organization, and similar difficulties were observed, including again the disenfranchising of Macintosh and older Netscape users, and also the use of weak encryption. Whereas in Arizona, the voters could use other polling sites for traditional balloting in the primary, the ICANN voters did not have this option. The voting method used a complicated iterative proportional recount process in order to avoid runoffs, which caused considerable confusion among those who were casting ballots. This election is currently under dispute, as the entire process was questionable.

Chapter 2

The ballot is stronger than the bullet.

-- Abraham Lincoln, 1856

2.0 Technology

Voters in the United States (and in democracies around the world) currently cast their election ballots in one of a variety of ways, only some of which involve electronic tabulation -- that is, the use of a computer-based system to tally the selections. The most current statistics available on voting equipment use are those collected by Election Data Services from secretaries of state, city, and county clerks.[BRA93] This information was tabulated for the following systems, in 1992 (numbers represent counties, and are thus not representative of machine quantities): Datavote punch-card 84, electronic 117, lever machine 795, mark-sense (also known as optical scanner) 627, paper ballot 662, other punch-card 651, and mixed systems 189. As these numbers reflect counties, they are not indicative of the number of voters, nor even of the quantity of systems deployed. Nevertheless, it is clear that paper-based voting (Datavote punch-card, mark-sense, other punch-card, and paper ballot) prevails, as used by 64% of the counties. Of the systems that tabulated by computers (Datavote punch-card, electronic, mark-sense, other punch-card), these were used by at least 47% of the counties. Interestingly, some 21% of counties that year still counted paper ballots by hand. By the year 2000, there was a strong trend away from hand-counted paper ballots and lever machines, and an increase in mark-sense and electronic systems. Table 1 and the sections that follow in this chapter compare and contrast these major voting methods.

Voting System	% 1980	% 1992	% 2000
Paper ballots	40.4	21.2	12.5
Lever machines	36.4	25.4	14.7
Punch-card	19.1	23.5	19.2
Mark-sense	0.8	20.1	40.2
Electronic	0.2	3.7	8.9
Mixed	3.0	6.1	4.4

1980 and 2000 county percentages from [CAL01], 1992 percentages from [BRA93]

Table 1 – Voting Systems by County

2.1 Lever Machines

These older-style mechanical machines, still in use in many localities and based on late-1800s patents by Thomas Alva Edison and others, do not have any computer interfacing for electronic tabulation, but it is useful to describe them here for contrasting and historical purposes. Essentially, an intricate system of gears and levers increments various individual tabulating units, which are similar to the mechanical odometers on many cars. These machines, some of which have been in operation for close to a half-century, are deteriorating and are becoming difficult to maintain. Although replacement parts can be machined indefinitely, highly skilled workers are required to keep the machines operating, and voters are not especially tolerant of downtimes occurring during elections.[BAQ90] Furthermore, the sheer weight of the equipment, and the related storage, transportation, and maintenance costs have caused a considerable percentage of the districts still using them (36% in 1980, close to 30% as of 1988 [FRE88], 25% by 1992 [BRA93], and under 15% by 2000 [CAL01]) to consider alternative methods.

It should be noted that the acceptance of the concept of a "machine" into which votes are cast does lend the communities that currently deploy the mechanical systems to consider other more modern (computer-based) balloting machines as acceptable substitutes. Indeed, voters who are unfamiliar with the operation of these mechanical systems often incorrectly surmise that their entire ballot image (rather than the incremental votes) is recorded within the machine and that the counter number on the side of the device (which in many communities matches the voter number they are given when they sign the register) provides the election workers (those people assigned to monitor the election at the polling place) with a way to reconstruct how a certain person actually voted. In fact, individual ballots can not be reconstructed in their entirety using lever machines (unless photographic or video monitoring methods are used), although certain other features of these devices do compromise some minor privacy issues. For example, it is possible for the nearby election officials to overhear the sound made when a metal window slider has been opened to perform a write-in, the identity of the voter may be remembered and then matched with the paper write-in strip at the end of the day (providing, interestingly, a type of covert channel). Since write-ins generally occur with relatively low frequency, this is not much of a concern.

One method for compromising lever machines, well-known to election insiders, involves deliberate physical tampering. It is possible to roll-back individual tabulating units so that they start out at a high number (such as 900 for a unit that counts to 999) prior to the beginning of the voting session. In order to make it appear that the units were all cleared when the machines are examined by poll workers at the start of the day, a small sticker with a 0 printed on it can be used to cover the improperly exposed 9. These stickers could be removed discretely after the election, when the machines are returned to storage. Using this example, a candidate whose tally was rigged would need to get 101 votes in order to register 001 (or 201 to register 101, etc.). Given prior knowledge of local voting patterns (information which is publicly available), it is possible to target certain precincts where such a candidate handicapping scheme would have a low chance of discovery while possibly contributing to an alteration of the overall election outcome. This particular method, along with overt tampering by jamming levers and damaging ballot faces on machines, was believed to have played a role in a Presidential primary in New York during the 1980s, and is understood to have lead to the City's consideration of an electronic replacement for their mechanical election systems.

The elaborate scheme described above involves access to the individual pieces of equipment both before and after the election, as well as possible collusion among election officials, which makes it difficult to accomplish. In general, though, the lever machines have proven to be reasonably reliable and secure. Most importantly, these devices provide simple methods whereby open auditing can be performed, within the election process, by relatively unskilled observers.

2.2 Paper Ballots

To date, tabulating systems for computerized vote counting have predominantly used punch-card and mark-sense ballots. Ballots are perforated or marked by the voters, usually within a small privacy booth, and then deposited into a sealed box. These formats are also often used for mail-in absentee ballots by voters who are unable to get to the polls. All paper balloting methods suffer from the problem that ballots can be lost, substituted or even duplicated with modifications, and deliberately or inadvertently damaged or voided. Punch-cards are more vulnerable to erroneous tabulation due to hole misalignment and "hanging chad," a situation in which the small bits of paper are not fully removed from the holes when entering a vote, and they fall back into the same or different holes as the cards are handled during the counting process.[DUG88] This occurs more often with pre-scored cards, and when the stylus used by the voters to release the chad is not spring-loaded (as recommended in Roy Saltman's 1988 report [SAL88]). Mark-sense ballots are similar to the type of forms used for multiple-choice exams, and their mechanical readers have a high error rate, some of which is also due to user error (or sloppiness in marking the selections).

Both types of paper systems do offer an anonymous ballot (if one does not look for fingerprints). Various municipalities use sequentially numbered ballots for accounting purposes, which (if not shuffled before issuing to voters) provide tracking possibilities if the number is recorded when a ballot is issued to a voter. Problems related to paper ballot substitution are discussed in Section 4.1. The existence of a physical ballot that the voter can handle and examine prior to dropping it into the ballot box does provide some assurance of auditability, via an electronic or even manual recount, if necessary. Voter feedback is further enhanced if the card has printed directly on it the names of the candidates, and the hole or mark is associated with the ones selected. (Note that minimal standards established by the Federal Election Commission (FEC) do not require that the candidates' names be printed on the cards.)

Although election officials have long known about problems related to the lack of self-verification of punch-cards, a simple fix involves the placement of a reading unit at each polling place which pre-scans each marked ballot, allowing the voter to privately determine whether the punches accurately reflected their intended choices. This has not typically been provided for voters using the punched-card systems (although such readers are occasionally available for the mark-sense ballot systems) largely because of additional cost and time factors.

2.3 Direct Recording Electronic

Still relatively new to the scene, Direct Recording Electronic (DRE) voting systems are gradually being introduced throughout the United States. These differ from the punch-card and mark-sense systems in the same way that lever machines differ from paper ballots. Essentially, the vote is collected through a series of selections on the face of the machine, and the tabulation is performed internally, within the device, as the election proceeds. No receipt or confirmation (other than a visual scan of the voting surface prior to the conclusion of the vote) is given to the voter. (The reason for the lack of receipts is explained later in Section 3.3.) This method is highly flawed, as Roy Saltman observed:

> "...the voter is given some reason to believe that the desired choices have been entered correctly into the temporary storage, but no independent proof can be provided to the voter that the choices have, in fact, been entered correctly for the purpose of summarizing these choices with all others to produce vote totals."[SAL88]

Election districts presently using lever machines find that DREs offer a number of advantages over the mechanical units. Lever machines can weigh up to seven-hundred and fifty pounds, and take considerably more space than do the two-hundred pound (or less) DREs, so storage and transportation costs are reduced. The voters find the DREs more comparable to the lever machines than the punch cards, and the expense of providing the cards is eliminated, although this cost is transferred to the reconfiguration of the DREs for each election.[TRO89a]

DREs are currently manufactured by a variety of firms, some of which were previously lever machine vendors. They differ in construction and features between manufacturers, and, within the same brand, also vary due to state and local requirements. A DRE is not just a simple stand-alone or networked personal computer system (although such units are now being proposed); rather, it is intended to be a secure unit with back-up and user accessibility features.

Generally, the machines contain the following components:

1. A panel whereby votes may be entered (using pressure-sensitive keys or a touch screen).
2. Some indicators (typically light-emitting diodes or screen displays) showing which selections have been made.
3. One or more central processing units that control the internal operation of the machine.
4. Memory circuits containing the object code of the software providing the machine's functionality.
5. Other integrated circuit chips and discrete components that are part of the hardware design (this may include I/O, interrupt, timer and other logic units, as well as such items as modems and printers for supplying the results).
6. A printed circuit board (or boards) on which the chips and components are installed and interconnected.
7. A power supply along with a battery back-up system in case of power loss.

8. Some mechanism for recording write-in votes (this may be an alphanumeric panel with a display unit, or a hand-writeable paper tape accessible through a window).
9. An operator's panel that permits verification that the machine appears to be functioning properly, and may also allow viewing of the vote tallies at the start and end of the election session.
10. A unit that records the votes entered (this may be a removable cartridge).
11. A tamper-proof case that houses the voting machine components.
12. A screen or curtain that permits the voter to use the machine in private.

Additionally, a separate electronic unit may reside at some central location for the purpose of collecting the individual machine tallies (for example, using cartridges removed from DREs following the closing of the polls, or via modem data transfer) and computing the election totals.

2.4 DREs versus Lever Machines

Aside from the obvious differences of electronic and mechanical modes of operation, DREs bear some similarity to lever machines. One might first think that Saltman's statement regarding the lack of proof that one's vote has been entered correctly also applies to lever machines, and in some sense it does. Certainly a gear or other component in a lever machine could slip and miss the tabulation of a single vote, or it could even jam or be rigged in such a way that an entire sequence of votes might be omitted. But in the case of a mechanical unit, the hardware of the entire machine could be carefully examined and signs of wear significant to cause slippage or jamming would be observable, and even repeatable -- but not necessarily deterministically.

With DREs, it is conceivable that a computer glitch or intermittent failure could happen in such a manner that would be undetectable, yet have an effect on the vote tabulation. Many DRE manufacturers have designed redundancy and error-checking into the circuitry used to collect the votes from the entry panel, and into the elements containing the vote tallies, so such intermittent malfunctions would likely be detected. Although it should be noted that computer self-checking is often a "fox guarding the henhouse" source of risks, since equipment that is broken or has been tampered with can not necessarily be relied upon for accurate reporting.

If we assume that DREs are constructed in such a manner that machine failures are always accurately reported, then we must direct our attention to the detection of deliberate acts on a DRE by an individual or group of individuals, which could "deny or abridge" (the phrase used in the U.S. Constitution) the voter's rights. In this regard, the lever and DRE machines are substantially different. With a lever machine, deliberate acts of tampering with the equipment are likely to leave some physical trail of evidence, and the correct operation of the device is observable and reproducible. Individuals can be trained to perform repairs, and replacement parts can be manufactured the locally, so that dependence on vendors of lever machines is unnecessary.[BAQ90] On the other hand, with a DRE, it is possible for an insider to affect the tabulations in a manner which may be undetectable, and irreproducible.

Certain of the components of the DRE systems (such as the CPU chips) must be outsourced, providing a serious vulnerability flaw, and other parts may not be obtainable from manufacturers other than the original vendor (due to copyright, patent, and trade secret restrictions placed on proprietary materials). Repairs may be difficult, if not impossible, unless specific detailed information (like circuit diagrams and even source code) is provided along with the electronic machines. It is often the case that the purchaser of a DRE must be required to establish a permanent ongoing relationship with the vendor, to maintain the proper working order of the machines. The vendor's agents may even have continued (and unsupervised) access to the machines' internals, in order to perform maintenance tasks. Such access provides opportunities for alterations of hardware and/or software either surreptitiously (thus compromising integrity) or benignly (through accidental introduction of defects into the system). This is discussed later in the thesis, in the section on misuse.

There is yet a further concern related to DREs, which is also not present in either the lever or paper-based voting systems. With all other systems, if an election is going to be "fixed" (the vote altered in favor of a certain candidate or candidates), it needs to be done the old-fashioned way, that is, one vote (or one machine) at a time. With the DREs, or any system that relies on a centrally programmed unit to tally the votes, an entire election can be "thrown" merely by affecting a pervasive change in the software. This could be inserted early in the development process, such that it would be unnoticed at election-time during routine validation procedures. Unintentional flaws in voting systems can, once discovered, also be globally exploited with nefarious intent. Once the proliferation of a specific vendor's products reach critical mass in the country, it is conceivable that a national election could then be tampered with electronically. (The manner in which this could be done in an undetectable fashion is described in Section 3.3 of this thesis.)

2.5 Vote-by-Phone and Internet Balloting

The Internet is not always a nice place, and its superhighway neighborhood continues to deteriorate annually in terms of crime and flagrant misuse, as illustrated in some chilling statistics published by U.S. News and World Report in August 2000:

> "Last year, the Federal Trade Commission received more than 18,000 Internet-related complaints. That's more than double the previous year's volume. For the first six months of this year, it received 11,000 complaints. The FBI opened 1,500 online child sex cases last year, up from 700 a year before. Businesses too, are feeling the pinch. According to a recent survey by the Computer Security Institute and the FBI, 70 percent of companies experienced cyberattacks in the past year, up from 42 percent in 1996. Nearly 300 companies reported losses of more than $265 million."[MAN00]

Especially relevant to voting is the dramatic increase in identity theft, estimated to affect over 500,000 victims in the year 2000 alone, and considered, by the FTC, to be the fastest growing U.S. crime. Yet, despite the fact that Internet and personal identification security appears to be at an all-time low, municipalities are forging (and two meanings of that word are perhaps apt) ahead in their efforts to certify Internet-based election systems often as proposed solutions for the low voter turnout problem. Lauren Wiener, in discussing the subject of erroneously automated processes, disagrees:

> "People don't care to vote, not because it's too hard to get to the polling place, but because they don't think their votes make any difference. Allowing people to vote from home by phone is not going to fix that. What *will* fix it is a profound change in the political system. This point was amply demonstrated by the 1992 U.S. presidential election, in which more eligible voters participated than in any election in decades, because people perceived that they had a real choice. Vote-by-phone schemes are a diversion."[WIE93]

Setting aside the implication that U.S. voters are either too disillusioned, too lazy, or too busy to take a few minutes to step into the polling place near their home to cast a ballot on election day, these methods do seem to offer the benefit of enhanced convenience and simplicity. Indeed, if the process is made simple enough, one will be certain to collect more ballots than there are registered voters or even citizens living in the municipality at the time of the election. Conversely, phone and Web voting can be made more confusing, rather than less, with nested sub-menus and doubt over whether one has actually cast a ballot in its entirety. Here, neither simplicity nor complexity are necessarily good things.

The Internet differs from a controlled local or even wide-area network in that it is globally accessible for transmission and reception of data to/from any and all other Internet connected devices. Internet security features are largely add-ons (authentication, firewalls, encryption) and problems are numerous (denial-of-service attacks, spoofing, monitoring).[WEI00] [BLA00] Hence, interfacing to the Internet could be, in itself, considered to constitute a security breach, in that wide attack and monitoring opportunities are provided that would not be possible with individual DRE kiosks, or in a closed network setting where all clients and servers are known and identified prior to system operation. The movement of information over the Internet involves routing through dynamically determined and difficult-to-trace paths, whereas a controlled network can establish and track data transmissions. The Internet includes systems that are not subject to local or United States laws, and whose operators can not be expected to comply with local voting regulations.

Systems on the Internet range from hand-held units to mainframe computers, created by a variety of hardware manufacturers, running different operating systems and browsers, all of which are making a 'best attempt' at forming non-WYSIWIG displays from the downloaded data. Certain Internet Service Providers also ship unsolicited advertising along with the requested Web pages, an action that would need to be suppressed during balloting in order to conform to campaigning restrictions at polling places. That such a diverse and distributed system could be relied upon to conform to locally imposed standards for a short-term application of a critical nature, but which is not of commercial merit, is contrary to the very spirit and current intent of the World Wide Web.

A further ramification of the off-site process is that it begins to blur the lines of the geographical communities upon which representation is based. In certain respects, this can be good -- for communication with the public, the chat-room concept can be converted into electronic town meetings, and elected officials can hold virtual sessions -- people need not be in the same physical space at the same time, although one must be certain that the voices of the technologically deprived continue to be heard, and that the dynamic of discussions is not adversely affected.[WIE93] But off-site balloting methods are a first step toward eliminating the community-based poll watching process, which is so essential in providing checks and balances in assuring that the voters are who they say they are; that they are voting only once and not casting ballots for other parties; that privacy is maintained and coercion is not occurring. These principles are embodied in the first three of Shamos' commandments. The remote vote opens the door for organizations to create their own convenient balloting locations and to consider intimidation of the members who don't use them, thus enhancing the role of special interest groups in determining the outcome of elections. Of course the same could happen with paper absentee ballots, but it is the trend toward off-site voting as the norm, rather than the exception, that is viewed as problematic.

A December 1999 Public Policy Institute of California statewide study (used by the California Internet Voting Task Force, see Section 6.3) indicated that the youngest group of voters (ages 18-24) now has the most access to Internet services (70%) and that there is a steep decay in interest in I-voting through the various age groups (to only 20% at age 65+), although these demographics may eventually shift. At present, though, I-voting threatens to create an Internet-savvy electronic elite who can cast ballots more easily than the elderly, poor, and possibly also physically challenged. The same would be true for remote voter registration.

The U.S. primaries have been viewed as a test for I-voting; as there is a general belief that since these elections are only run-offs within each party, the need for bipartisan poll watching is moot. However, this belief is untrue. As demonstrated in the 2000 U.S. presidential primaries, significant impact on the choice of candidates can occur through cross-party voting, thus enabling the opposing party to eliminate a person whom they may deem hard to beat.

Off-site balloting also opens the door for vote-selling, a powerful way to throw an election to the highest bidder, clearly unintended by election officials hoping merely to modernize their systems. An Internet site named voteauction.com, purported to be an academic project by James Baumgartner, an MFA student at Rensselaer Polytechnic Institute, appeared in August of 2000 and began collecting data from citizens interested in marketing their votes.[AND00a] Following publicity by WiReD News, and subsequent threats of legal action for violation of New York State election laws, it was sold and reopened from an off-shore location where prosecution of the site's owners may be able to be circumvented.[AND00b] Those who offered their ballots for sale, or even only inquired about doing so, may still be subject to criminal consequences, including fines and loss of right to suffrage -- if states with applicable laws choose to pursue the matter.

Internet voting vendors on the scene include the previously mentioned VoteHere.net and (the currently most prominent) election.com, as well as other new dotcom companies. Election.com, of Garden City, New York, describes itself in its promotional literature as providing "complete election services for public and private institutions" and claims to be "the only global Internet election management company committed to making democracy work better throughout the world." Their services include election planning, migration assistance, ballot design and production, election notification, tracking and tabulation, and archiving. The matter of whether a single company should be permitted to provide all of those services for the same municipality is itself a risks issue. As we have learned that the software industry, in general, is not necessarily best served by monopolies (although Microsoft™ might disagree), so too, the democratic process could be easily compromised or held hostage by a lone powerful company or consortium in the absence of any real competition. Given the recent rise and fall phenomenon in dotcoms, one should also be skeptical of doing business with voting system vendors who may not have the ability nor the intention to service their products or customers for the long haul.

Bizarrely, even after problems occurred with the earlier-noted Arizona primary, the vendor's new literature made the following (wholly erroneous) claims: "All election.com security solutions comply with rigorous industry standards and are compatible with the widest range of platforms available. There is no additional software for a voter to download or install, eliminating the threat of virus or any elements that could compromise an election." Additionally they promise, but have not offered any proof that "election.com guarantees security during all phases of the online voting process, including member authentication, session verification and server protection. All election.com servers are protected against attack by state-of-the-art firewall and intrusion detection software." Election.com has continued its permeation into the election process by announcing in June 2000, that all U.S. voter registration, change of address, and absentee ballot requests could be processed from their Web site, hence opening the door to massive disenfranchisement of voters whose identities may be more easily co-opted on-line.

A host of additional related issues pertaining to user verification, auditing, denials of service, firewall breaches and so forth are critical to I-voting [NEU00a] and these points are discussed at some length later in this thesis.

2.6 Voting System Comparisons

Table 2 summarizes the major issues with each of the different voting systems. Ongoing costs have been noted since these are often overlooked at the time of system procurement. Verifying refers to the ability of the voter to independently (visually) examine the ballot to determine if it has been prepared properly, prior to casting. Overvoting occurs when a ballot has more candidates selected in a race than permitted, and undervoting involves skipping a race entirely or voting for fewer candidates in a race than allowed. (Although punch-card and mark-sense systems do not inherently have capabilities for alerting voters to overvoting or undervoting, the addition of a scanning mechanism at the polling station would permit such checking to be performed.) Overvoting generally results in voiding of all candidates chosen for that particular race, whereas with undervoting the choice to skip a race does not detract from any vote selected – in both cases, the voter may be unaware that they have done this, so additional feedback is important. Recounting involves the ability to perform a retabulation of the ballots cast. Full-face ballots are required by some municipalities, and are also believed to reduce undervoting for lower-rank offices. Pros and cons address other matters of general concern.

Table 2 does not differentiate between DRE and other forms of off-site voting (as mentioned in Section 2.5) since the Internet and vote-by-phone systems would also involve direct recording of ballot images. All of these types (whether used remotely or at polling places) suffer from the same flaw -- the voter can not independently verify that the ballot cast corresponds to that being recorded or tabulated -- without introducing mechanisms that would encourage vote selling. The DREs offer a slight advantage in that they could be adapted to produce voter-verified paper ballots (this will be described in Section 3.3) for recount purposes. Remote voting reduces voter authentication assurances, and increases denial of service attacks. Self-contained (non-networked) DREs may be somewhat easier to secure than distributed systems operating on a variety of platforms, but this does not eliminate other security flaws (such as Trojan horse attacks) that are possible in all of the computer-based voting systems.

Devices > *Issues* V	**Manual Paper Ballots**	**Lever Machines**	**Punch-Card**	**Mark-Sense (Optical)**	**Direct Recording Electronic**
Ongoing Costs	Printing of ballots	Setup	Cards and ballot books	Printing of ballots	Recording device, setup
Vote Verifying	Yes	Yes (somewhat)	Usually No	Usually Yes	No
Overvote	Possible	Impossible	Could warn	Could warn	Preventable
Undervote	Possible	Possible	Could warn	Could warn	Could warn
Recount	Manual	Observation of totals	Re-scanned or manual	Re-scanned or manual	Via recorded internal trail
Full-Face Ballots	Yes	Yes	Usually No	Usually Yes	Possible with some
Pros	Printing can be in-house, paper audit trail	Mechanism can be viewed and audited	Provides paper audit trail	Easy to use, paper audit trail	Rapid tally
Cons	Counting is slow, labor-intensive	Bulky and no-longer built	Chad may complicate interpretation	Voters may make some errors, cost	Expensive, audit not independent

Table 2 – Voting System Issues

Chapter 3

One man shall have one vote.
-- John Cartwright, 1780

3.0 Sociotechnology

Technology alone does not eliminate or enhance the possibility of election tampering; it merely changes the platform on which the tampering may occur. The voters and the election boards who serve them must be aware of the risks of adopting electronic vote-tallying systems, insisting that the checks and balances inherent to the democratic process be maintained. This is typically done through legislation, verification, validation, monitoring, and auditing.

3.1 Legislation

Voting is a fundamental right of the citizens in a democracy. The subject of denial or abridgment of voting rights appears in five of the twenty-six amendments to the United States Constitution.[US] As important as it is, though, the U.S. government leaves the method of administration of elections as a matter of states' rights. Each state can, within the parameters of the Constitution, decide who is eligible to vote (based on residency requirements, felony convictions, and so on). As well, every state is permitted considerable latitude in establishing vote tabulation regulations. For example, in some states a person may leave the voting booth without voting for any candidate, but in others, a blank ballot might be deemed invalid. Some states provide a single lever (or check-box on a paper ballot) whereby a slate of candidates in a particular party can all be selected together in a general election -- other states do not allow such mechanisms for same-party voting. Some municipalities use proportional or preferential balloting (typically local, like school boards), where multiple choices are permitted and candidates attaining over a certain percentage of votes cast can move on to a further run-off election. In such an election it would be reasonable to tally more votes in a race than there are voters casting ballots, so the use of simple totals would not provide a verification check. Each state, therefore, has established an extensive set of laws pertaining to the election process. It is important to recognize that even though voting machines are used nationwide in federal elections (Presidential and Congressional), the states (and local municipalities within them) independently set the methods for approval, use, and inspection of their own equipment. Hence, the machines themselves may differ, even between municipalities within the same state, in order to comply with the various sets of regulations. This further complicates the verification and validation process (as described later).

In the area of electronic voting systems (punch-card, mark-sense and direct recording), the FEC in 1990 released a set of "minimum performance, testing and security requirements that can be voluntarily adopted by state and local governments for voting systems in their jurisdictions." [FEC90a] The operative word here is *voluntary*, because at present the FEC does not require, nor has every state adopted, this voluminous set of standards. Furthermore, the standards have been criticized as having been highly influenced by particular voting machine vendors, so there is some question as to the rigor with which they were prepared. Since, from the time of the 1988 Presidential election, over 50% of the votes cast have been tabulated by some form of computer system [FRE88], and since this form of tallying may increase as old lever machines are replaced, the verifiability of the results is a matter of growing concern.

Proponents of electronic voting systems say that sufficient controls are being exercised, so that attempts to subvert an election would be detectable. Yet numerous computer security experts have pointed out that the FEC's voluntary voting system standards may not be adequate to ensure election integrity. Numerous incidents of electronic voting difficulties have come to the attention of the press, yet these cases typically are dismissed (often due to lack of proper evidence, resulting in part from inadequate audit trails) and to date there have been no convictions for vote fraud by computer.

Presently the resources of the National Computer Security Center (NCSC) and the National Institute for Standards and Technology (NIST) have been (for unknown reasons) largely ignored by the federal, state, and local agencies responsible for overseeing the U.S. election process. The expertise of these organizations, and others in the private sector who have extensive experience in all phases of computer security, should be involved throughout the processes of regulation, procurement, development, and certification of all systems used in elections. Existing programs, such as the Common Criteria, and its predecessor, the Department of Defense's Trusted Computer System Evaluation Criteria, should be applied to election equipment.[CCI99][NCS85] [NCS90][NCS92] It is incumbent upon the federal government (through Congress and the Federal Election Commission) to take a leadership role in establishing and mandating minimal compliance standards, not just suggested guidelines. Government officials, at all levels, must work actively with vendors and municipalities to guarantee that elections are carried out with the highest integrity that technology will allow.

3.2 Verification and Validation

The verification of correct operation of mechanical hardware at the level of complexity of a lever machine is certainly tractable. On the other hand, verification that an arbitrary piece of software (running on a von Neumann architecture computational device) performs a certain task, is known to be intractable (except in some limited applications). This issue is compounded when dealing with secure systems, as Fred Cohen writes:

> "The general facilities exist for providing provably correct protection schemes, but they depend upon a security policy that is effective against the types of attacks being carried out. Even some quite simple protection systems cannot be proven 'safe'. Protection from denial of services requires the detection of halting programs which is well known to be

> undecidable. The problem of precisely marking information flow within a system has been shown to be NP-complete. The use of guards for the passing of untrustworthy information between users has been examined, but in general depends on the ability to prove program correctness which is well-known to be NP-complete."[COH84]

Validation of other voting machine aspects can be shown to be NP-complete, for example, keypress combination testing is analogous to CNF-satisfiability.[BAA88] Since correct operation is not provable, examiners are required to resort to weaker forms of verification. How this is done is, as mentioned earlier, left up to the states to decide.

Established quality assurance (QA) methods for computer system verification and validation have long been in place for most government contract work. The standards vary in accordance with the level of QA necessary to certify a product for use. A criticality level is assigned to components -- the highest level is usually reserved for those whose failure could result in a potential loss of life. As our elected officials have the power to declare war, one might want to use no less care in designing voting equipment, so the highest level of QA should be used in the examination process. In accordance with this understanding, the verification and validation procedure should (at least) include examinations of:

1. The vendor's System Requirements Document (SRD) in conjunction with the state election laws in order to ascertain conformance.

2. The System Design Document (SDD) used by the vendor to create the voting machine. This should be compared with the SRD.
3. The System Quality Assurance (SQA) plan used by the vendor to establish test policies, procedures and practices, reviews and audits, configuration management, security policies, archiving, and supplier control, in the manufacturing process.
4. The System Verification Plan (SVP) produced by the vendor, as well as those used by independent test agencies.

The FEC voluntary standards seek to address this level of QA, but are somewhat weak, especially in the areas that would be covered by the System Verification Plan. Here, the Common Criteria could be used to guide the evaluation of the system documentation and resulting product, since its philosophy is one of assurance based on active investigation. This investigation need not be excessive, since one of the goals of the CC is to apply the least amount of effort necessary to provide the required assurance. Increased effort can be used, in examination as required, on scope (extent of the system), depth (finer level of design and implementation detail), and rigor (application in a structured, formal manner). Further discussion of the Common Criteria, and its evaluation process, is in Chapter 5. Another view of security evaluation considerations can be found in the Generally Accepted System Security Principles (GASSP), but since these have not become a government standard, the CC is more appropriate for voting system use.[ISF97]

System verification involves both black box (e.g. input) and white box (e.g. path) inspections. Well-documented studies reveal that code walk-throughs can detect 30% - 70% of software errors, many of which can not be found through input testing alone.[MEY79] The FEC's reluctance to recommend code examination possibly stems from the proprietary nature of the system source code. Presently, the vendors have been reluctant to reveal the details of their software -- none have voluntarily provided a complete document to any independent agency for an in-depth white box examination. In the most extensive study to date, the lengthy New York City procurement process did result in an evaluation of the Sequoia Pacific target machine during the mid-1990's, but there was not sufficient time or funding provided for a complete software review. Nor was the examiner (SRI International) a wholly independent agency, having being paid by the purchaser with severe constraints imposed on the process by the vendor. Besides, even though source code inspection (under nondisclosure agreements) showed no security holes, the study enumerated many ways in which the election process could nevertheless be compromised.

The reasons for this evasiveness is illustrated in this comment by a Vice President of DFM Associates, a company producing vote-counting software, speaking against source code deposits:

> "Our feeling is these people [the escrow agents] will read everybody's software....and eventually they will filter into the business and steal our ideas."[TRO89b]

In a competitive market, it is understandable that vendors would want to protect the code that operates their DREs under the cover of trade secrecy, but it may not be to the advantage of the voters to allow such protection. Certainly the vendors would be entitled to copyright and patent privileges where applicable, and these should provide enough legal grounds to secure their property. Of far greater concern is the implication that having knowledge of the software would provide a road map to rigging an election, and this is more likely an unstated reason for code privacy by the vendors. Without a complete set of source code, though, it is unlikely that any DRE could be verified and validated to current software QA standards. This provides a strong argument for Open Source Software, discussed in Section 3.4.

Yet, the source code alone is inadequate for providing a complete internal system verification. The compiler used to generate the object code must be available, and all hardware specifications must be revealed, down to the chip level. Even this may not be sufficient -- Ken Thompson, in his now-classic Turing Award Lecture, demonstrated how a Trojan horse could be inserted into the compiler itself. He summed this up quite simply when he said:

> "You can't trust code that you did not 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."[THO84]

This provides insight into the statement by Robert Boram, chief engineer of DRE voting machines at R. F. Shoup, on rigging elections despite the existence of internal ballot images:

> "I could write a routine inside the system that not only changes the election outcome, but also changes the images to conform to it." [DUG92]

As well, Penelope Bonsall, director of the National Clearinghouse on Election Administration, a branch of the FEC, said:

> "Sure, anything is vulnerable to fraud or manipulation. But you've got to have technical knowledge and you've got to have collusion. One person can't do it in most systems."...."The feeling in the industry is that there are so many easier ways to affect an election that tampering with the tabulating software doesn't really make sense."[TRO89b]

But just because tampering with the software may not be the easiest method does not mean that it has not or will not be done. Thompson's implication is that the hooks and backdoors, particularly those within compilers and operating systems, exist and have already been proliferated invisibly throughout the industry. Under this view, software rigging is assumed to have already happened, rather than just a speculative possibility. One could extend these assumptions as well to the hardware. Presently there is nothing that restricts vendors from using custom integrated circuit chips in the DREs, and some do, even for the CPUs. It is not inconceivable that a crafty individual could devise a set of microcoded instructions that would be activated only under certain situations. Reliance on any particular vendor or brand of components would therefore increase vulnerability. Some chips now even permit internally reconfigurable microcode as well as microarchitecture, and such a self-modifying CPU could erase any trace of its own subroutines once they were executed. With election dates and times being well-known and predictable, this could occur within the space of microseconds during the voting session. To again quote Bonsall: "You've got to trust somebody, somewhere."[TRO89b]

Yet the procedures used for verification and validation appear to abdicate the lion's share of trust to the vendors, by testing systems only to the functional level. Providing thorough source code and circuit examinations for DREs in each state, and furthermore in each municipality requiring variations of machines within each state, is a Herculean task -- one that is likely not to be affordable, even if it were accomplishable.

More recently, in commenting about Internet voting, Bonsall said: "There are many, many, many, many issues."[WOL00] But the FEC has not made any effort to either clarify or do anything about these issues. What is desperately needed is the formulation of a methodology whereby each municipality can at least confirm that the code and circuitry contained in each of its own machines is identical to that verified by the state or other independent testing authority. One could suggest that it might be expedient for the

FEC (or even NIST) to be relied on to validate all voting machines, but unless each of the states voluntarily agrees to submit their equipment to a national checking authority, the issue of states' rights in monitoring elections would be impinged.

3.3 Audit Trails

If we place our trust in the vendors (who do, after all, have the most to lose if their DREs are considered unreliable or compromisable), we would be remiss if we did not provide some method, beyond functional testing, to ensure that vote entries are properly tallied. Numerous individuals and organizations have looked to audit trails to provide the necessary (albeit not sufficient) verification.[SAL88][FEC90]

The U.S. election process, by its very nature, is both secret and adversarial. The (essentially) bipartisan system provides an analogue to the checks and balances established by the branches of government. In the polling place, representatives of each party oversee the voting activities, such as inspecting the machines to see that they begin with zero vote counts, and insuring that the totals are recorded properly on the returns at the end of the day. (Let us ignore, momentarily, the inequities that can arise when minority inspectors are appointed by members of the majority party -- this is discussed in Section 4.1.) With a DRE, indeed it is possible for the machine to print or display any arbitrary value that does not necessarily reflect the true vote tallies. Since results are stored electronically, examination of the tabulating units is not possible. Our checking system, therefore, becomes largely procedural rather than validatory.

Audit trails seem to provide an alternative partial means of validating the DRE tallies. Each voter's entire ballot image could be recorded, and printed out at some later time. The vote could then be hand-tabulated and confirmed. It might even be reasonable to require that this be done in some small percentage of precincts after each election. Here the secrecy of the vote comes into question. If the machine were to record the ballots in the sequence they were voted, it would then be possible to determine an individual voter's ballot, by numbering each person as they enter the machine. The vendors who propose using such a system also provide for randomization (and even further encryption) of ballots within the recording unit, in an effort to make individual ballot identification unlikely. However, typical seeded computer functions only provide pseudo-randomization, and encryption (discussed in Section 3.5) should not be relied upon as a secure solution, so this is only a partial fix.

When using a DRE, one's ballot is intended to be private, and no receipt should be issued to the voter. Ballot contents receipts are problematic, since voters would be responsible for their disposal, and their existence might encourage unscrupulous candidates to exchange votes for gold ("turn in your receipt after the election"). Similar problems arise if the voter is issued an encrypted code number which can be used to "look up" their ballot following the election. With banking, one does want to be able to be identified with one's account (unless it is in Switzerland) -- whereas with voting, it is reasonable to insist that an individual should not be able to identify any specific ballot from the group, so that one's right to anonymity may not be compromised. The enigma, referred to in Section 1.2, is that the audit trail printout, without any form of verification that particular ballots correspond to those that were actually cast, actually provides no more security than does a fox guarding the hen house.

One might believe that the functional verification process, in issuing input sequences that are examined against the audit trail printout, guarantees that the auditing method is correct. It does not take much imagination to create a scenario in which a particular Trojan horse program is activated by a special sequence of keypresses (which can even include simultaneous multiple keypresses) on the DRE (the systems allow for a voter to select a candidate, and then de-select it, numerous times -- the ballot is not actually recorded until the end of vote is signaled by a final keypress or lever). This Trojan horse might then increment one candidate's tally by some percentage, decrement the opponent by the same percentage, and then reconstruct the ballot images to make them appear legitimate. (If the ballot-recording device is write-once, modifications could occur during the voting session while subsequent votes were cast, rather than after the fact.) To further evade detection, the keypress sequence might be distributed over a series of votes. As the number of keys on some DREs exceeds five hundred, it is unlikely that validation testing would inadvertently discover the fraudulent trigger sequence. Again, one could require that the audit trail transcribe the entire series of keypresses entered by each voter, but unless the order of the ballots was also retained (which would violate the privacy of the vote), the sequence might be hard to reconstruct.

A common criticism of the above scenario is that collusion among a group of individuals would be necessary in order to carry out this scheme, and that this would encourage detection. Far less cooperative effort would be required for a greater effect using the DREs than with O'Neill's paper ballot switching (described in Section 4.1) or with other less sophisticated techniques that have been widely used without repercussions. The collusion may not happen precisely in the manner outlined here, a wide variety of possibilities are certainly plausible.

Perhaps the only reasonable method of auditing a DRE election would involve printing each ballot before the voter exited the machine. The voter would then examine the ballot (through a photo-resistant viewing screen) for correctness and then authorize its placement into a sealed ballot box (the paper could drop over a series of small bars to provide randomization). Voters would immediately register a protest if the printout did not concur with the display on the DRE (a procedure would need to be in place to deal with malfunctions -- likely similar to that which occurs with lever machines, such as providing emergency paper ballots to voters -- until the equipment has been returned to proper operational status). Once the printout had been examined and confirmed, the voter would press a key to clear the computer display and would then exit the voting booth.

The primary difficulty that one could envision with such a system (other than the additional cost and time) would be paper jams, although advances in printer technology make this only a marginal concern. There could also be some user error, such as forgetting to authorize placement of the paper ballot in the box. Even though printers have been successfully used on ATM machines for many years, and the voting system vendors have been made aware of this audit-trail solution, none (to date) have provided it in their products. One could speculate that they do not choose to do so because the paper would be able to expose electronic vote tabulation flaws (or rigging), lowering confidence in their products.

At the end of the voting session, the machine totals could be provided for early returns. Candidates could opt to concede on the basis of those reports, or could wait for the certified tallies, obtained by optically scanning the paper ballots (using multiple systems for checks and balances, as discussed later in Section 5.2.L on recounts). Hand-counting could even be used (perhaps only as a preliminary check at a random selection of precincts), and the results compared with the machine totals. All of the paper ballot boxes must be impounded and secured for use in the case of a recount. The paper method at least provides some way of returning the verification process to the voters and multi-partisan election overseers. (See paragraphs 11-15 in the Appendix of this thesis for a discussion of the necessity of human-performed recounts with punch-card ballots.) The availability of different recount methods provides additional assurances as each are successively applied. As long as it is agreed in advance that a hand-tally of the paper (printed) ballots always provides the final arbiter for election results, the issues involving computer-based vote tallying fraud can be rendered moot.

3.4 Open Source Software

The Open Source Software (OSS) movement has been viewed by some as offering a potential solution to certain aspects of the vote tabulation problem -- specifically, the code secrecy and software validation issues.[PER97] Although parts of the movement have maintained a somewhat nefarious pose, with their Robin Hood-like "give it all away" attitude, this concept has, regardless, achieved growing legitimacy through the widespread distribution of powerful tools (such as Linux and other GNU packages from groups like the Free Software Foundation) into the computing community. DARPA is now seriously promoting this alternative to closed source, proprietary software.

Neumann expresses the pros and cons of OSS (or as he calls it, "open-box") rather succinctly in the following statement:

> "The benefits of nonproprietary open-box software include the ability of outside good guys to carry out peer reviews, add new functionality, identify flaws, and fix them rapidly -- for example, through collaborative efforts involving people widely dispersed around the world. Of course, the risks include increased opportunities for evil-doers to discover flaws that can be exploited, or to insert trap doors and Trojan horses into the code."[NEU00b]

The Open Source concept pertains to the availability of the program code for perusal and possible modification. The use or proliferation of Open Source products may still be encumbered by licensing fees, or available for unpaid distribution, at the discretion of the original issuers. Only the object code may be available, and there may be restrictions or warnings regarding use. For example, the Free Software Foundation provides modifiable, redistributable, royalty-free source code, but no guarantees are given regarding fitness of purpose.

The argument has been successfully made and demonstrated that businesses can survive and indeed thrive with a base in Open or Free Software, so the voting machine vendors should not necessarily fear a loss of revenue by adopting this modality. (Indeed, a large part of the voting system business involves after-sales support such as election setup, maintenance contracts, supplies, etc., so a "give away the razors and sell the blades" philosophy is certainly viable as long as source code availability does not adversely impact security or operations.) What remains is to assess the merit of such an approach for vote tabulation.

Say, then, that open source vote-counting software is produced and provided for full examination by anyone who cares to peruse it. The system needs to be designed in such fashion that intimate knowledge of its inner workings does not produce (as mentioned earlier) a road map to throwing the election. Individuals and groups would be given ample opportunity to examine (or even modify or create) the code and deem it to be functionally correct and free from operational compromises. As discussed before, since software correctness proofs are infeasible except in the most simplistic cases, the best that can be expected is a partial code certification. Does this mean that the systems deployed to use this code are therefore fine? Not in the least. As noted by Thompson and described in Section 3.2, even if one has verified that the source being compiled is the properly certified version, one must also examine the compiler used to create the running object code, the compiler that compiled the compiler, and the entire system upon which the code is being installed. If all of this was also provided in an open format, can we still trust the system? No, because there need to be assurances that all of the various components previously examined were indeed the ones used within the system during the election (which could be done using cryptographic checksums). We further need to secure all parts of the system before, during, and after the election so as to maintain integrity throughout the entire process. And we need to do this for all components, at all election locations, throughout all of the aforementioned times. This becomes a strategic nightmare, if one intends to continue the practice of dropping voting machines off (at schools, firehouses, libraries, etc.) days before an election and picking them up days later, especially if these machines contain code that is accessible for modification. Open source requires, indeed demands, constant vigilance. Thus, maintaining a machine-style election is therefore infeasible if we want to use open source.

With a paper-based election, though, it becomes plausible to use open source just for the vote tabulation process, and such has often been proposed, yet no implementation appears to be forthcoming. The code must not be intertwined with an elaborate hardware mechanism, so that it can be compiled and run on different platforms, thus reducing the possibility of a hardware-based "fix." In such a format, the only validation necessary would be on those components that are performing the actual post-election vote tabulation. With paper ballots available for recount, it is conceivable that multiple versions of the open system could be used to check the tallies, in parallel, by independent poll-watching groups. Here again, as in the case of audit trails, paper offers a clear and obvious solution -- use hand-marked (or computer-generated but human-readable) ballots, let the citizens certify various open tabulation systems, run multiple tally passes, and provide a procedure for reconciliation in the event that different results are produced. But this has not been deemed acceptable, primarily because, in the view of many election boards, paper ballot systems are obsolete and cumbersome. Other downsides of paper systems are discussed in Sections 2.2 and 4.1.

3.5 Encryption

Encryption masquerades as a technological (algorithmic) solution to a sociological (privacy) problem, and encourages false assurances from developers, legislators, and others who would promote such products. This is a bold statement, yet one that is supported by the growing realization within the computer industry that cryptography does not, in itself, provide a secure system solution. The well-known cryptographer, Bruce Schneier, recently recanted his earlier belief that "it is insufficient to protect ourselves with laws; we need to protect ourselves with mathematics." He now asserts that cryptography is no 'magic bullet', stating:

> "...I have made a living as a cryptography consultant: designing and analyzing security systems. To my initial surprise, I found that the weak points had nothing to do with the mathematics. They were in the hardware, the software, the networks, and the people. Beautiful pieces of mathematics were made irrelevant through bad programming, a lousy operating system, or someone's bad password choice."[SCH00]

Schneier paraphrases a famous Roger Needham and Butler Lampson quotation thusly: "If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology." Cryptography is only a 'middle' solution -- even with perfect mathematics, the data must exist in an unencrypted format at the endpoints (before and after encryption). There, the security vulnerabilities are high, and sociological as well as technological issues are apparent.

Cryptographic solutions are intertwined with many security aspects of voting systems, and are noted numerous times within the Common Criteria (specifically in discussion and implementation of trusted channel mechanisms, data encryption and/or decryption, digital signature generation and/or verification, checksum generation for integrity and/or verification of checksum, secure hash, cryptographic key encryption/decryption, cryptographic key agreement, export and import of data with associated security attributes, and unobservability of resource use). Yet the Common Criteria essentially punts on the issue of cryptography by focusing on the administrative tasks affiliated with key generation, distribution, access, destruction methods, and auditing of applicable roles and operations, while leaving the implementation to the system developer, as long as the standards used are specified.

Assurance that encryption algorithms are selected, implemented, and functioning properly is left to the evaluator, with only the following admonishment: "The subject of criteria for the assessment of the inherent qualities of cryptographic algorithms is not covered in the CC. Should independent assessment of mathematical properties of cryptography embedded in a Target of Evaluation be required, the evaluation scheme under which the CC is applied must make provision for such assessments." Spoofing, bypass, deception, trapdoor or other malicious circumvention of an entire crypto system is possible -- for example, a user could be redirected to code which produces the appearance of an encrypted message through simple bit shifts rather than actual cryptographic algorithms -- so assurances for these components is essential, in order to maintain integrity. Since the environments (operating systems, compilers, etc.) upon which the crypto systems rely are inherently weak, vast security holes persist.

Even if an assessor chooses to assume that the mathematical implementation, and its system embodiment, of an established cryptographic algorithm is correct (an assumption which could have serious consequences if proven false), the standards themselves are questionable. Crypto-hacking has been likened to the upward spiral of an arms race, as noted in a recent IEEE Computer article:

> "More powerful and less expensive computer technology, such as faster microprocessors and improved distributed-networking techniques, makes it easier and less costly for criminals to use a computer or groups of computers for brute-force attacks on encryption algorithms. As the computer technology used in hacking improves, there is a steady demand for better security.[CLA00]

Time to break encryption is typically measured in MIPS-years, but prior estimates have been deemed erroneous when parallel techniques were employed on networked processors. The 56-bit Data Encryption Standard, in common use since 1977, was recently broken in under 24 hours using a distributed-Internet technique. The traditional fix has been to increase key size, as in the Advanced Encryption Standard, which supports 128, 192 and 256 bit keys, but this also increases encryption/decryption time requirements. The RSA method, which is computationally intensive, involving prime-number factors of large integers, has been cracked for some increasingly long public keys, using number-field sieve algorithms. A new standard, Elliptic Curve Cryptography, has a smaller key size and shows promise, but possibly only because the mathematicians have not had enough experience with cracking it yet. The advent of ultra-high-speed cellular CPUs in the next decade, as silicon reaches its density limitations for IC's, could rapidly outstrip today's MIPS-year estimates making all current crypto standards obsolete. Furthermore, distributed cracks make hardware speeds essentially irrelevant. So, although cryptography may mask data content from casual perusal, permanent assurance of privacy should not be assumed.

Thus, it is possible that an approved encryption solution designed into a voting system now may be outmoded by the time it is deployed or within its projected usage lifetime, necessitating re-tooling. Even more certainly, ballots encrypted using a current-day standard, which are collected and archived as part of the audit trail or for use in contested election recounts, may be able to be decomposed at some later date without any access to the voting software. This creates the potential for revealing the identities of remotely authenticated voters along with the contents of their ballots cast. Such illicit decomposition and distribution of voting information may be regulated by laws, but again, if the encrypted ballots are broadcast over the Internet, such laws may not necessarily apply to all intercepting parties.

Further problems with cryptography are based on the use of keys (as found in smart-card devices and other methods used for authorized access), which are subject to misuse by theft, voluntary disclosure, or crypto-hacking. If one is provided with a key, the manner in which it was distributed must be scrutinized. If a key is generated by the user, whether internal or external to the secured voting system, it must not be able to be replicated or transmitted without the user's knowledge. In the absence of biometric authentication, someone with access to the voter list could scrutinize it at the end of the voting session for 'no-shows' and use the remaining keys to cast additional ballots, for the purpose of throwing the election. It is even conceivable that an unscrupulous vendor, hoping to demonstrate popular acceptance of a voting system, could use some of the keys to cast ballots just to increase the apparent turnout.

Yet another issue involves the association of voter authentication information with ballot data. Using the metaphor of sealed, paper absentee ballots mailed inside of signed envelopes, say that an encrypted ballot was transmitted along with an 'outer wrapper' of an encrypted voter identification tag. The voter ID could first be revealed and verified, and then the encrypted ballot sent to another area in the system for decryption. One must guarantee that the entire package is not retained for later decryption as a whole, and also that there is no way to associate the decrypted names with the ballots. Even if this were possible (and it may be difficult to assure), one is still left with the problem of assuring that the decrypted ballot accurately reflects the voter's choices (and this is unresolvable, as discussed earlier).

3.6 Access Controls

Authentication, authorization, and access controls are the triple-A of system security. *Authorization* involves the placement of restrictions on the people, processes, times, and purposes of system use in order to form a blockade against attack and misuse. Authorization should apply to voters as well as system administrators and operators (election officials), and also to developers and repair personnel. Authorization is also necessary for secure intersystem and intrasystem communications. An important aspect of authorization requires ensuring that trapdoors, Trojan horses, and other forms of circumvention are not and can not be present. The usefulness of authorization is diminished without the addition of *authentication* in order to confirm that identities are indeed genuine. *Access control* pertains to every aspect and stage of the voting system, from design and development of hardware and software, through testing and deployment, election use, vote tallying, auditing and recount.[NEU95]

The triple-As must be applied for all users of the vote tabulation system, with their extent dependent on the amount of containment (or lack thereof) under consideration (for example, an Internet-based system would require more stringent controls than would an isolated tabulation unit). For the voters, these aspects would be enforced differently for the various voting system styles. Using paper-based systems, authorization and authentication takes place at the time of ballot issuance, and the voter's only access is to their own ballot data that will later be recorded. Voters using DREs or other consoles within a polling place setting would similarly be authorized and authenticated by the poll workers, who may even perform a short "set-up" procedure to clear and ready the electronic ballot prior to each voter's use, as is currently done with lever machines (through controls on the outside of the unit). Here, it would be important that poll workers be properly authorized and authenticated, but this can be performed at the beginning of the entire voting session, as long as the voter can not access the set-up functions. Note that with an interactive system, the manner of enforcement may require some additional hardware that only the poll workers are allowed to use. With off-site balloting, authorization and authentication becomes more complex. California proposes to do this through a signed paper request and returned passcode exchange process. Voters logging-in to a remote system should be precluded from performing or accessing operations, functions, and data unrelated to the balloting process. This may be secured through hardware segmentation -- for example, the balloting software may be accessible only through a separate server than the one used for tabulation.

A considerable portion of the Common Criteria is concerned with the triple-As. The discussion in this Section and its subsections is consistent with the information on this subject found in CC Part 2.[CCI99] The method used by the CC assigns attributes to users (or user groups), subjects, information, and objects in order to provide general information or specific access control content for security enforcement. The scope of access control provided by functions within the system, as well as the amount of interaction permitted, and even the permission to establish a session, are all based on the attributes supplied at the time of use. These attributes must be clearly defined. The importance of this aspect can not be underestimated, since the CC states that: "the unambiguous identification of authorized users and the correct association of security attributes with users and subjects is critical to the enforcement of the intended security policies." "This involves not only establishing the claimed identity of each user, but also verifying that each user is indeed who he/she claims to be. This is achieved by requiring users to provide some information that is known by the [system] to be associated with the user in question." Data itself may need to be authenticated, in order to guarantee its validity and that information content has not been forged or fraudulently modified (data issues are addressed in subsection 3.6.2). Attributes may be bound to subjects (which can be processes or data) that act in a user's behalf; such binding may need to be maintained through various stages of operation. Security attributes may be subject to expiration based on time limits specified by an authorized user or via management functions. The list of security attributes that can expire, the roles that can modify them, and the actions to be taken on expiration, should be specified.

The CC devotes specific attention to the subject of user authentication. It suggests that the types of user authentication mechanisms supported and the required attributes on which these mechanisms must be based should be defined. These comprise the actions performed on behalf of the user before the claimed identity is authenticated and "should have no security concerns with users incorrectly identifying themselves prior to being authenticated." Unforgeable authentication in this family addresses mechanisms providing protection of the authentication data, and "provide confidence that users authenticated ... are actually who they claim to be." Since it is impossible to detect or prevent sharing of passwords outside of system control, this "may be useful only with authentication mechanisms that are based on authentication data that cannot be shared (e.g. biometrics)." Users may be allowed to perform certain actions prior to authentication, or all actions may be prohibited until authenticated, depending on system requirements. It should be noted that different mechanisms may be used for the authentication of different specific events. Forged or copied authentication data may be detected and prevented from use. Some authentication data may be single-use only (this could be applied to individual voters with new authentications issued for each election), and re-authentication may be necessary for subsequent access. In remote voting, the issue of re-authentication in the case where a voter logs-in but fails to complete a voting transaction must be addressed. Re-authentication of a user at defined points in time can be necessary, and the rules and conditions for such should be specified. The feedback provided to the user during each of the authentication processes should be defined, including indication of typed and masked characters, as well as other prompts, so the voter can follow an anticipated validation sequence.

Access control policy, says the CC, "is based upon the concept of arbitrary controls on the interaction of subjects and objects. The scope and purpose of the controls is based upon the attributes of the accessor (subject), the attributes of the container being accessed (object), the actions (operations) and any associated access rules." The access control mechanisms should be identified, along with the definition of triplets of subject, object and operation sets covered. Components may be iterated multiple times to different subsets of operations and objects to specify more than one access control policy. Subset access control "specifies that the policy cover some well-defined set of operations on some subset of the objects." Complete access control requires that all possible operations on objects are covered and also that the designer demonstrate that each combination of objects and subjects is covered. Access control may be performed by functions (rules) that implement the security policy, governing access among controlled subjects and objects using controlled operations that state when access is granted or denied, and rules to explicitly authorize or deny access based on a privilege vector associated with a subject.

Access control may also require that the system be capable of terminating a session after a specified number of unsuccessful authentication attempts within a period of time. This blocking may include disabling an individual account, or the point of entry (the terminal or its address). The invalid authentication attempts need not be consecutive in order to trigger the locking mechanism. An administrator alarm is typically sent. At least one account must not be deactivated, so that total denial of service can be prevented. Automatic lockouts will persist until re-establishment conditions have been achieved -- restoration to the normal state may need administrator involvement. Although repeated entry attempts may be a signal of attack, the blocking thresholds may need to be set somewhat high to accommodate unsophisticated voters who make typing and other entry errors. For administrative functions, controls could be more stringent. Auditing may be provided to track unsuccessful as well as successful uses of the authentication mechanisms (although the latter creates a privacy conflict when applied to voter access), detection of fraudulent authentication data, reuse of authentication data, failures, checks, and results. Revocation may also be applied, or session establishment constrained, based on other usage patterns, time periods, location, and security attributes (such as identity, security clearance level, integrity level, membership in a role or group). Resources subject to access revocation, and the rules and attributes used for application, should be managed closely. Management of session establishment conditions and auditing of session establishment attempts, denial, and access parameters should be provided.

Various security roles may be specified in order to manage system functions. Modification of security roles (such as those that can invoke access revocation) must be closely managed and audited. As well, any modification of the behavior of critical system functions, even by authorized users, should also be subject to audit. Certain users may also have permission to modify data, and this should be audited, especially in critical areas. It would not be unreasonable to deny all permission to modify particular data segments, such as ballot contents. The assignment of roles must also be managed, since there may be additional rules that control relationships between roles, and some roles may not be allowed to be assumed by the same individual. Auditing may involve modifications to the group of users in a role, the uses of the rights of the role, unsuccessful attempts to use a role, and explicit requests to assume a role.

Access may also pertain to deterrence and resistance to unauthorized physical modification or substitution, physical tampering and interference. Monitoring components may be included in order to passively and unambiguously detect or actively resist such tampering, along with required responses to provide automatic notification of tampering attempts. The user role that gets informed in the case of intrusion, the devices that perform the notification, and the automatic responses to tampering, may all require management. Auditing of physical intrusion detection should be performed.

Firewalls are but one of many preventative measures that can be employed to assist with access control. These can be used to provide a barrier against unwanted system entry, as well as an alert mechanism for the outward dissemination of confidential material. For example, firewalls can provide some defense against passwords traversing communication lines in the clear, a subversion avenue which can easily be picked up by packet sniffing software. Although firewalls are not a complete solution (breaches are noted even with the best systems), they can be a useful component in a larger, overall access protection scheme.

3.6.1 Trust

Trust is closely tied to the subject of access control. Indeed, in the security field, the word 'trusted' is often applied as a modifier to 'access' or 'channel' or 'system' (as in the title of the DoD TCSEC standard). The actual meaning of trust, though, is both difficult to define and an elusive quality. In the context of providing access to a system, say via passwords, trust is granted to the user, and it is expected that the user (as well as the system) will maintain the provided secret. The CC gives the example where "a trusted channel mechanism that relies on cryptography to preserve the confidentiality of information being transmitted via the channel can only be as strong as the method used to keep the cryptographic keys secret from unauthorized disclosure." Assuming that secrecy is properly maintained, the user must then also demonstrate trustworthiness, in the form of actions consistent with what is expected for the accessed application, or as expected from prior user activity. Thus (unlike the stock market), past performance *is* (or should be) a good predictor of future behavior.

The concept of a *reference monitor*, first described in a 1972 study for the United States Air Force, has formed the basis for trusted systems since that time. The reference monitor concept, and its embodiment as a reference validation mechanism, is used to enforce the access control policies of a secure system. The three design requirements specified for a reference validation mechanism are that it must be:

1. tamperproof;
2. always invoked (that is, not bypassable); and
3. simple enough that it can be subjected to thorough analysis and testing, with an assurance of completion.[NCS85]

The CC addresses the 'always invoked' aspect of reference mediation by stating that protection against unauthorized operation must ensure that *all* enforceable actions requested by untrusted subjects be validated by the monitoring system before succeeding. Bypass of monitoring would be permitted only for certain 'trusted subjects' that would have to be authorized by other procedures for identity validation prior to being granted such special privileges. The system must be designed to require that enforcement functions always be invoked and succeed, prior to proceeding with any subsequent secured function.

Automated mechanisms can be provided that analyze system activity and audit data, looking for possible or real security violations. The CC states: "this analysis may work in support of intrusion detection, or automatic response to an imminent security violation." In the voting scenario, independent vote collecting units would not likely contain such sophisticated analysis tools (beyond perhaps just anomaly and fraud detection), although the central tabulating unit, especially if networked for voting data transmission, should be required to have this analysis in force.

The CC defines various forms of violation analysis:

1. Simple violation threshold detection via accumulation or combination of defined auditable events.
2. Maintenance of system usage profiles representing the historical patterns of usage performed by members of the profile target group.
3. Detection and reporting of certain signature event occurrences that are known to pose threats to security enforcement.
4. Multi-step intrusion representation and detection through comparison of "system events (possibly performed by multiple individuals) against event sequences known to represent entire intrusion scenarios."

With regard to usage profiles, the CC elaborates: "Each profile represents the expected patterns of usage performed by members of the profile target group. This pattern may be based on past use (historical patterns) or on normal use for users of similar target groups (expected behaviour)." "Each member of a profile target group is assigned an individual suspicion rating that represents how well that member's current activity corresponds to the established patterns of usage represented in the profile." Anomalous activities sufficient to exceed a threshold suspicion rating would trigger reporting by the system. There may also be anomaly detection tools, sets of auditable events along with rules to perform the violation analysis, specific lists of activities to be monitored, administrative notification along with the specification of conditions under which reporting should occur, and definitions on how to interpret suspicion ratings. Since a voting system should be applied only to certain specific tasks by its permitted users, monitoring targeted to these areas may be sufficient, as long as circumvention of controls is also prevented. Voter monitoring would occur during the procedures prior to actual ballot casting, so as to not interfere with anonymity. Voters access the system infrequently, so perusal of individual usage patterns is probably only relevant for administrators. One noted problem with this concept is that "anomalous activity gets integrated into the profile just like non-anomalous activity (assuming the tool is monitoring those actions). Things that may have appeared anomalous four months ago, might over time become the norm (and vice-versa) as the user's work duties change," so this would be important to consider when monitoring the administrative users who could gradually perform extraneous actions in an effort to eventually thwart the system at a critical time.

In the notes on violation analysis in the CC, it is stated that "in practice, it is at best rare when an analysis tool can detect with certainty when a security violation is imminent. However, there do exist some system events that are so significant that they are always worthy of independent review." Examples given include: deletion of a key security data file such as a password file; and a remote user attempting to gain administrative privilege. Such occurrences are known as "signature events" and, separate from any other system activity, can be viewed as having intrusion intent which may be of significant consequence in terms of violation potential. The CC further allows for the development of "a base set list of sequences of system events whose occurrence are representative of known penetration scenarios." These monitoring techniques should be used in voting systems.

3.6.2 Data

The ballot data is the most critical asset of the voting system, and as such, must be strictly controlled. Control should be applied to data collection, dissemination, modification, storage, use (for tabulation, audit, or other purposes), and maintenance. If the voting system also performs voter authorization tasks, this data would be another essential asset. The separation of ballot data from originator identity is a complex issue, highly relevant to remote voter authentication (this subject will be addressed at length in Chapter 5). The establishment of the voting system security environment must take into account the protection of all salient data assets (as well as other trusted system components and their data, of course). Various aspects with relevance to voting data are itemized below. These were abstracted from the numerous Common Criteria components addressing data topics.

1. *Data authentication* involves the functions required to generate all voting data authenticity as well as verification of the provider of such authenticity.

2. *Non-repudiation of origin* ensures that the originator of information is bound to the data in such way that it is not possible to successfully deny the data transmission. (This ensures that a person can not later claim that they didn't have an opportunity to vote, if their ballot was recorded as having been cast.) Evidence of origin is necessary, along with an association of the originator with the information, and transmission of this relationship to the information recipient or to a third party. The evidence of origin should be verifiable and should not be forgeable (this is extremely difficult to guarantee, and poses huge problems in the voting setting). The proof of origin can be selective, by specifying who can request evidence of origin, receipt, and/or verification. Certain conditions may be required to be met in order to be allowed to verify the validity of the evidence. It may be the case that information about the recipient must not be exported, in order to allow identity protection. The time of transmission, as well as the information for which the evidence applies may also be provided.

3. *Non-repudiation of receipt* ensures that the information recipient cannot successfully deny having been delivered the data. This typically requires that the subject who transmits information is provided with evidence of receipt. The evidence can then be later verified. Non-repudiation of receipt does not reveal the contents of the data transmitted, so it is feasible for use in anonymous voting. Receipt only pertains to delivery, hence it does not necessarily imply that the

information was read, used, or otherwise processed. Nor does it alone guarantee that transmission or recording of the data correctly preserved its contents.

4. *Data export* rules provide limitation on the outside transmission of certain voting information via communication channels or on transportable media such as diskettes or CD-ROMs. It may require that the data be encrypted, or that security attributes be transmitted along with the data in an accurate and unambiguous fashion, and it may also restrict the exportation only to certain designated sites.

5. *Data import* rules deal with the introduction of data from outside of the secured system, in such fashion that any associated data security attributes can be preserved. Mechanisms are defined for determining and interpreting associated security attributes in an unambiguous fashion. This would pertain to the use of Internet/Web forms of vote transmission, where voter authorization and ballot data could be provided from or via non-secure systems. Two possibilities for imported user data exist -- either it is accurately and unambiguously associated with reliable security attributes, or no (reliable) security attributes are available from the import source. Association may be by physical means (on the same media) or logical means (via checksums or other computation). This may be performed automatically or with human intervention. If no security attributes are present, they may be supplied by a trusted path or channel. If security attributes

are present, the consistency of the data must be ensured and rules and limitations on import must be specified.

6. *Information flow control* policies and functions establish requirements that address the common information flow issues, as well as addressing illicit information flows (such as covert channels, discussed at the bottom of this section). It is suggested that all information flows and operations pertaining to voter data should be covered by at least one information flow control policy. Enforcement may go beyond traditional access control policies (which could be insufficient if only data containers are dealt with and not the information itself), by identifying and describing non-interference policies and state transitions, as well as triplets of subjects, information and operations under control of the policies. For each action causing information flow, there will be a set of rules defining whether that action is allowed. Simple security attributes may be applied to explicitly authorize and deny an information flow based on attribute values, relationships between subject and security attributes, and rules to be applied. Attributes can be expanded into a hierarchy that can be used to implement a privilege policy through the use of ordering relationships. Attributes pertaining to the information may or may not remain with the information as it flows, depending on requirements, but the accessor may not change the attribute data (unless explicitly authorized to do so).

7. *Internal data transfer* addresses protection of voting data being transmitted across internal channels. Maintenance of separation of system data from voting data may be required. This aspect also deals with monitoring for detection of disclosure, modification, substitution, re-ordering, deletion and other data integrity errors. Actions should be specified to be taken if such error detection occurs. Automated recovery, such as via checksums, can be required. Loss of data availability may also be a concern. It should not be assumed that because the transfer is internal, that no threats to the data exist -- security attributes may still need to be applied and checked. Separate logical or physical channels can be required for protected use.

8. *External data transfer* addresses matters of confidentiality, security, and integrity of voting data being transmitted across external channels. Security attributes, encryption, spread spectrum techniques, authorization and other mechanisms may be applied during transmission in order to increase confidence, especially where full control of the channel is not possible. Confidentiality requires ensuring that externally transmitted data is protected from unauthorized disclosure between endpoints, while in transit. The policies enforced to decide who can exchange data and which data can be exchanged should be specified, along with whether the policy application is to a sending or receiving mechanism. Integrity involves the application of rules for protecting data during transmission from unauthorized modification. The ability to detect and possibly

also correct data modification may be included, along with notification to proper authorities when detection of modification occurs. Data deletion should also be detected, and recovery mechanisms should be available for use.

9. *Data availability* deals with the rules provided for prevention of loss of availability of voting data moving on an external channel. This is especially necessary when the voting system has distributed data components. Rectification of attempts to block data transmission should be considered. The types of data which must be available, and the conditions under which availability must be ensured (such as access to voter registration data by the authentication process) should be specified.

10. *Residual information protection* requires ensuring that deleted information is no longer accessible, and that newly created data areas do not contain information from previously used data spaces that should not be available. This would pertain to the data which links specific voters to their ballot images, for example, or to any ballot images which may remain on the casting machine. Logically deleted or released information, particularly on reusable resources (where deletion does not necessarily equate with destruction) is dealt with here. Objects stored off-line could be additionally problematic and may need to be addressed. If no residual information is to be allowed, then deallocation invokes the residual information protection function for the specified objects (or for all objects if such

is required). It should be noted that residual information protection during data deallocation may conflict with rollback operations that may be needed for information recovery in the event of system failure.

11. *Stored data integrity* refers to information contained in memory or on a storage device, and is intended to provide mechanisms to detect and possibly also recover from unintentional errors (such as hardware glitches). The types of integrity errors detected, the actions to be taken in case an error is detected, and the data attributes used as the basis for monitoring should be specified. This would be extremely important for the ballot data set.

12. *Data consistency* involves requirements for uniform interpretation of data and attributes of data shared by different processes. This may be performed automatically through application of lists of interpretation rules, or by using a pre-established exchange protocol. Consistency is also a concern with data replication, and it should be ensured system-wide, so that duplication of data does not introduce variation. This could occur, for example, if a channel between parts of the system became inoperative. Transaction logging and rollback could be used to perform replay operations, but such logging for voter data may need to be precluded. Areas of data which require replication and interpretation consistency should be specified.

13. *Time stamps* are a particular type of data that may have salient use by various processes (such as for audit and security attribute expiration, as well as voting time ranges), so it is important that these be correct and secure. Management of the time function so that it is reliably defined and constructed, along with auditing of any changes to the time is advised.

Covert channels (not to be confused with secure channels, see Section 3.7), are defined as "any communication channel that can be exploited by a process to transfer information in a manner that violates the system's security policy."[NCS85] There are various ways in which information can be secreted for later use. These are typically segmented into storage channels (which "allow the direct or indirect writing of a storage location by one process and the direct or indirect reading of it by another") and timing channels (which "allow one process to signal information to another process by modulating its own use of system resources in such a way that the change in response time observed by the second process would provide information").

The CC notes that "the issues concerning illicit information flows are, in some sense, orthogonal to the rest of an information flow control [security policy]. By their nature they circumvent the information flow control resulting in a violation of the policy. As such, they require special functions to either limit or prevent their occurrence." The CC deals with covert channels by addressing: rules that must be enforced and how security attributes are derived; hierarchical security attributes; partial or full elimination of illicit information flows; and monitoring of illicit information flows for specified and maximum capacities. Illicit information flows may be required to be entirely or partially eliminated, or just monitored for capacity and/or type. The entire elimination of illicit flows may not be possible, because of the resultant impact on the functional operation of the system. The CC Evaluation Assurance Level 5 is the lowest which requires covert channel analysis (and it is also required in all subsequently higher levels).

3.7 Secure Channels

For distributed systems, the communication channels between parts of the system and remote products that are interfaced need to be considered. A secure channel involves the establishment of trusted communications paths between users and the system, as well as between the system and other trusted information products. To paraphrase the CC, the trusted path "is constructed using internal and external communications channels (as appropriate) that isolate an identified subset of data and commands from the remainder of the system and user data. Use of the communications path may be initiated by the user and/or the system (as appropriate). The communications path is capable of providing assurance that the user is communicating with the correct system, and that the system is communicating with the correct user. In this paradigm, a trusted channel is a communication channel that may be initiated by either side of the channel, and provides non-repudiation characteristics with respect to the channel. A trusted path provides a means for users to perform functions through an assured direct interaction with the system." This is typically used for initial identification and/or authorization, but may be needed at other times in a user's session. "User responses via the trusted path are guaranteed from modification by or disclosure to untrusted applications." In a distributed (for example Internet) voting application, the trusted path scenario would likely be essential.

A trusted channel should be included whenever secure communication between the system and other remote trusted products is needed. Assured identification of endpoints and protection of channel data from modification or disclosure would need to be performed. (Although it should be noted that in a remote voting setting, endpoint assurances may be difficult to obtain, or may disclose voter identity information.) Configuration of the actions that require the trusted channel should be managed. Auditing can be performed for attempted uses of the trusted channel, failure of the trusted channel functions, and identification of the initiator and target. A trusted path is similar to a trusted channel, in that it establishes and maintains trusted communication between users and the system for security-relevant interaction, but an external channel is not involved. The management and auditing functions are also similar.

The rules for the creation and maintenance of a trusted channel or path connection for the performance of security critical operations should be specified. The possible initiators of the channel (the system, the remote trusted product, and/or its users, or all of these) should be stated, along with a list of functions that require trusted communication.

Secure paths and channels have been proffered as an approach for dealing with secure communication. Confidence is established that a user is communicating directly with the system, and that untrusted applications cannot intercept or modify the user's response. "Absence of a trusted path may allow breaches of accountability or access control in environments where untrusted applications are used. These applications can intercept user-private information, such as passwords, and use it to impersonate other users. As a consequence, responsibility for any system actions cannot be reliably assigned to an accountable entity. Also, these applications could output erroneous information on an unsuspecting user's display, resulting in subsequent user actions that may be erroneous and may lead to a security breach."

An example of a secure channel protocol is the Secure Socket Layer (SSL) used by the Netscape Internet browser and others. SSL uses a handshake method to establish communication between a server and a client that are authenticated to each other. They then negotiate an encryption algorithm and pass cryptographic keys prior to transmission of data.[FRE96] Data authenticity is provided through cryptography and digital signatures. One problem with SSL is that the server can be spoofed, with a hacked site on the other end pretending to be the real server, collecting passwords and other data, and even letting the voter think that a real voting session is occurring. Another problem is that transmission over the Internet is often not direct between the client and server, providing opportunities for monitoring, data collection, potential re-routing, and in-the-middle attacks. SSL authentication is thought of as end-to-end, but the middle is essentially unprotected.

Chapter 4

I always voted at my party's call and never thought of thinking for myself at all.
-- W. S. Gilbert, 1878

4.0 Sociology

Sociological issues, which may appear "light" in a computer science setting, often provide insight on system security and design matters, and hence are important to discuss briefly here. Additionally, the voter's perception of whether his or her ballot actually "counts" is a major factor in turnout on election day, and this is certainly a function of numerous sociological factors. This chapter addresses the business aspects of elections, and describes misuse techniques that can be employed to subvert the vote tabulation process.

4.1 The Election Business

In recent years, "getting out the vote" has become a billion-dollar industry. With Congressional races often costing in excess of a million dollars per candidate (an immense war chest is required just to lose a federal election), and state and local politicians finding that it takes tens of thousands of dollars to run even the smallest campaign, huge sums of money are changing hands long before the citizens make their way to the polls. Campaign costs include printing, rallies, postage, office expenses, TV and radio advertising, roadside signs, travel, mailing lists, etc., all in an effort to create an image for the candidate, and encourage his or her supporters to cast a ballot on election day. Large organizations, with particular agendas (such as the environment, civil rights, education, religion, labor, etc.) and powerful lobbies (representing factions from industry as well as non-profits), are now openly in the election business too, spending record amounts both in direct support of particular candidates, and more generally, in increasing public awareness of issues. The focus of all of this attention and expense, is that single day in November, when candidates are separated into winners and losers.

Election day brings with it a flurry of activity. Bipartisan officials are stationed at the polls to oversee the voting process. Committee people, ward leaders, campaign staff members and interested private citizens station themselves on the streets within the legal distance from the voting area, to distribute leaflets and talk to the voters. In many municipalities, law enforcement personnel (police, marshals) are on hand to discourage or deal with violations of election laws. At the end of the voting session, poll watchers (often appointed by the candidates) arrive to ensure, to the best of their abilities, that the votes are tabulated properly. Stringers from the press collect the tallies and phone in the results. Most of these individuals are paid, albeit often nominally, for their services in the election process.

The vote tabulation procedure itself incurs considerable expenditures. Whatever the method used to collect the vote (human or machine-readable paper, lever machines, direct recording computers) it must be prepared, distributed to the polling sites, tabulated, and collected after the election. Ongoing costs amortized over all elections may comprise all or some of the following: voter registration, procurement (evaluation and purchase) of voting systems, storage of voting machines and other related paraphernalia, and repairs. Costs directly related to an individual election include ballot printing, set-up (which may involve software programming), examination of equipment (before and after the voting session), and transportation of materials. If an election is contested there may be additional expenses related to court proceedings and recounts.

The entire election process is adversarial. It is not a "win-win" business game; most certainly some people are going to come out as losers. The overriding assumption is that each individual involved in the process is operating with some overt or possibly even hidden agenda, and wishes to tip the balance in favor of their agenda somehow. The poll workers who are closest to the voters are not "non-partisan" citizens working for the interests of the general public, rather they are declared to be members of one party or another. The system is supposed to provide "checks and balances" so that each group monitors the other, but in actuality this is often not the case. Workers from the "majority" party typically appoint or approve the workers from the "minority" party, who (in overwhelmingly majority districts) are frequently persons sympathetic to the majority ticket who have registered as minority party members for the sake of the appointment. Friends make sure their friends are employed for the day and, as often occurs in industry, most jobs are filled through inside contacts. Indeed, the process of becoming appointed as a poll worker is typically so obscure or confusing (with petitions that may need to be filed and deadlines that have to be met) that only insiders understand what must be done.

By the time the ballots are cast, each voter has been subjected to advertising and media blitzes, run the gauntlet of leafleteers, and has somehow sorted through it enough to make a decision. This decision may even be to let someone else decide, and paper ballots provide a wide avenue for fraud, as described by Speaker of the U.S. House of Representatives O'Neill in his autobiography (published after his retirement):

> The old-timers used to tell stories of how Martin [Lomasney, a Boston politician] would greet them at the polls on election day. "Here's your ballot," he'd say, "I've already marked it for you. When you get in there, pick up the ballot they give you and give them back this one." When you came out you'd give Martin the clean ballot, and he'd mark it off and give it to the next guy in line.[ONE87]

It could be conjectured that this simplistic but efficient procedure may be one of the reasons that the majority of ballots (even those counted by computers) are still cast on some form of paper in this country (around 64% of counties). Banks (as well as taprooms) used to be closed on election day, ostensibly so that people would not be tempted to "buy" or "sell" votes, but as these days of electronic tellers make cash available on a 24-hour basis, most now opt to remain open, unless constrained by local or state election laws. Clearly, in a paper election, it is possible to purchase votes on a person-by-person basis, and to verify that the voter cast the paid-for ballot.

We are led to believe that the adversarial process of elections somehow ensures a fair and accurate tabulation of votes. Instead, when it comes to elections, free enterprise is more often the name of the game. Consistent with this observation are the policies and statements by certain individuals and organizations who are major players in the vote tabulation business. Here are some press quotes on this subject:

> "Election system vendors are often forced by competitive bidding pressures to offer jurisdictions the cheapest possible systems, and the products they offer do not maximize fraud protection." [BUR85]
>
> C. A. Rundell, Jr., chairman and CEO of Cronus (a company that claimed to have about 40% of the election-service market in the mid 1980's), replying to a journalist's question regarding jurisdictional use for their equipment -- "We certainly are not going to provide you with a list of customers and the kinds of systems they have." "We've got to ask how much competitive intelligence we divulge to our competition." [DUG88]

Interestingly, the above comments were made in the 1980's -- since then, not much has changed -- no one is making any attempt to hide the fact that vote tabulation is a business, that elections can be rigged, and that votes can be bought. Highly-paid teams of consultants are brought in to city and state Boards of Elections to place their imprimatur on the procurement of new, unauditable, and tamper-prone voting systems, despite warnings from scientists, engineers, and spokespersons from the National Institute for Standards and Technologies, the Federal Election Commission, bar associations and public interest groups, as well as occasionally even the vendors themselves.

If it looks like a business, sounds like a business, and smells like a business, then the business of elections will, more than likely, continue to operate under the traditional law of supply and demand. Suppliers will provide systems that attempt to meet election criteria specified by the purchasers, and when such criteria is lacking, they may use their own judgment in manufacturing these products. If it is truly the case that the voters (or those who purchase voting systems) desire that the process remain unsecured, systems that allow opportunities to throw elections will continue to be created and sold. Given the huge sums of money that are spent on election campaigns these days, it is not implausible that such flawed systems would then be targeted for subversion. Alternatively, if the public demands that their election systems be subject to such regulations and scrutiny as those applied to other businesses that rely on the public trust (such as banking or health care), then rigorous standards and compliance procedures can begin to be developed and enforced.

4.2 Misuse

Peter Neumann and Donn Parker define a set of classes of computer misuse techniques [NEU89], each of which applies directly to the electronic vote tabulating scenario:

External misuse pertains to the observation or theft of information relating to the voting system. It might involve rummaging through discarded printouts, monitoring systems via their radio frequency emission patterns, or visually obtaining a password (by watching the keystrokes of an operator). Passwords, although ubiquitous in the computer industry, are inherently risky, easy to capture and reuse. External misuse actions are generally passive, but the information obtained may later be applied to a more overt system misuse or attack.

Hardware misuse can be either passive or active. Passive actions could include the placement of a data collection unit within the voting system, or obtaining a discarded ballot cartridge for the purposes of reverse-engineering a cloned device. Active misuse includes theft of systems or components, intentional physical damage (dropping of equipment, slashing of ballot faces, insertion of glue in keys or switches, dousing with liquids, etc.), modifications (to allow improper access), power supply tampering, and interference (magnetic, electrical, etc.).

Masquerading is deliberate impersonation in order to obtain information or gain access to the system. Individuals might pose as service personnel or as authorized operators before, during or after an election. In this way, they can collect passwords, tamper with hardware, software and data, or directly manipulate the programming and tabulating processes. Vendors and election boards may also inadvertently employ double-agents from competing equipment vendors, or other persons with adverse hidden agendas.

Subsequent misuses can be set up through the use of software Trojan horses that are time-triggered (so that they do not appear in pre- and post-election testing), or input-triggered (through the appearance of a particular data, command, or even ballot sequence). Code for such misuse can be written to "self-destruct" following execution so that it does not appear in later system audits. Source code escrows can be rendered useless by involving the compilation or assembly process in performing the actual Trojan horse insertion. [THO84]

Controls established within the system for security and auditability may be bypassed both intentionally and accidentally. Exploitation of design flaws in multi-user systems, by using installed trapdoor programs, may enable unauthorized access to election software and data by individuals logged in through separate accounts. Password attacks can be used to obtain "superuser" system status, from which audit trails can be turned off or modified to remove traces of system penetration.

Authority status may be misused actively within the system by legitimate superusers as well as by those who are masquerading as such. Some of these misuses that apply to voting systems include: alteration of data, false data entry, and denials of service. These superusers could even selectively apply misuse techniques such that they occur when known opposition members are in the voting booth, if they are able to perform hacks in real time.

Passive misuse of resources can include browsing of data, global searching for patterns, covert channel exploitation, and access to groups of files that can be used collectively in a more powerful way than when used separately. Within the voting context, the information gathered in this manner can generate statistics that could be used in subsequent attacks on the same system, or on other systems located elsewhere. System-specific information, such as ballot cartridge programming or vote tabulation methods, can be transmitted to other municipalities that have similar installations, for use in subverting elections.

A particular type of passive misuse of resource involves having direct access to individual ballots, vote totals, population statistics, registration information, and preexisting voting patterns. It is possible for employees of election companies who provide full service operations to have access to all of these databases simultaneously. This information could then be applied in order to shift tallies in swing precincts in subtle ways that would be hard to detect. This is extremely powerful, since many elections are won by small percentages, and even just a handful of votes can be critical for local races. One would certainly not add votes to an overwhelmingly Republican district such that a Democratic candidate would win (or vice versa), since that would likely raise eyebrows. Targeting areas which are more evenly balanced demographically, on the other hand, would probably be overlooked, but could still result in altering the outcome of an election. In this way, a vote system vendor whose products have reached critical mass throughout the country (this might only need to be 5% or even less), could silently affect the results of a national election. Performed on an ongoing basis, this could gradually shift the representation of the larger elected bodies (Senate, Congress) dramatically. If one intended to subvert the democratic process, this might be an effective technique.

The lack of timely intervention in the event of a detected or potential problem could also be viewed as a form of misuse. This can include inappropriate disposal or handling of election and computer media, non-reporting of an observed system attack, or other breaches of policy and procedure. Here a 'cover-up' to save face in light of a system problem may be considered a further improper system use.

System tampering can also be indirectly applied to other criminal acts or fraud. This form of misuse would enable illegitimately elected individuals to later perform illegal activities involving misuse of their fraudulently obtained powers, such as inappropriate bidding for contracts, misuse of funds, or nepotism in hiring.

Denials of service, such as those produced through active misuse techniques or time-triggered Trojan horses described above, pose a major problem for elections, since voters must be provided with a method to cast ballots on a specific day, within a certain period of time. Voters who are told that the "system is down" may be unable to return to the polling place at a later time (or log back in, in the case of Internet voting), and are thus disenfranchised. Certain municipalities could even be targeted for denial of service in order to shift result totals. The time-critical nature of election day, along with the fact that the general election always occurs on the first Tuesday of November each year, makes this form of attack particularly likely, yet this potential problem has not been sufficiently addressed by the manufacturers and purchasers of electronic vote tabulation systems.

It has been asserted (by industry and government representatives, as earlier quoted) that collusion would likely be necessary in order to tamper with an electronically tabulated election. The above list of points of attack indicates that collusion is not necessary as many forms of system invasion can be performed by a single individual. As audit controls for access and use of vote tabulation systems have typically been lax or nonexistent, the attack can be done in a straightforward manner, often with minimal technical skills or knowledge. Such attacks may be motivated by politics, monetary rewards, power, foreign agencies, business interests, and subversion, to name but a few.

Chapter 5

As soon as questions of will or decision or reason or choice of action arise, human science is at a loss.

-- Noam Chomsky, 1978

5.0 Common Criteria

The Common Criteria (CC) anticipates vulnerabilities of Information Technology (IT) products through failures in requirements, construction, and operation. It establishes a set of seven Evaluation Assurance Levels (EALs) through the association of various features with appropriate analysis methods, and provides a structure for the generation of Protection Profiles (PPs) for the Targets of Evaluation (TOEs). Assurance, and subsequent system verification and validation, are thus established through extensive and methodical examination. (Quotations throughout Chapter 5 are taken directly from CC Parts 1, 2 and 3.)

The Common Criteria replaces its predecessors, the Trusted Computer System Evaluation Criteria (TCSEC) [NCS85], and TCSEC's international superset, the Information Technology Security Evaluation Criteria (ITSEC), as the methodology of choice for certifying secure systems. The CC and TCSEC/ITSEC evaluation standards differ considerably in their approach. Whereas TCSEC/ITSEC could be viewed as 'bottom-up' in that security classes are defined in terms of sets of features that must be present and examined, the CC is 'top-down' since it focuses on reducing the risk of undesired behaviors through testing, design review, and implementation.

The CC assists in establishing security objectives for systems targeted for evaluation. The specified objectives are then refined into a set of requirements, in order to ensure the intended security goals. Requirements can be functional (such as identification, authentication, and audit) or can provide assurance. Assurance is expressed by degrees, as defined by EAL selection, and is derived through assessment of correct implementation of the security functions, and evaluation in order to obtain confidence in their effectiveness. Application of the CC does not constitute a proof of correctness, as it is noted that security objectives "generally include both requirements for the presence of desired behavior and ... the absence of undesired behavior. It is normally possible to demonstrate, by use or testing, the presence of the desired behavior. It is not always possible to perform a conclusive demonstration of absence of undesired behavior." Recalling the earlier discussion in Section 3.2, even the presence of the desired behavior can not necessarily be assured through demonstration alone. Both situations pose problems for voting system implementation.

It is important to again mention that neither the CC nor TCSEC/ITSEC has been mandated or voluntarily complied with in the vote tabulation arena. Yet, there is universal agreement outside of the voting community (by the Department of Defense, some health care organizations, as well as the CC's U.S. administrative body -- the National Institute of Standards and Technology) that the CC presently provides a reasonable, state-of-the-art benchmark for security assurance. Hence, exhibiting the configurations of various electronic vote tabulation systems at specific EALs, while simultaneously showing each to be incapable of satisfying the collection (or certain subsets) of Shamos' voting constraints, would demonstrate the inability of such resultant systems (even if evaluated properly) to be appropriate for their purposes as designed. Such an approach would not require actual system specifications; rather, it would cite essential components and procedures, and identify their inherent weaknesses.

The CC disclaims its own comprehensiveness, saying that "the CC and the associated functional requirements described herein are not meant to be a definitive answer to all the problems of IT security. Rather, the CC offers a set of well understood security functional requirements that can be used to create trusted products or systems reflecting the needs of the market." But the CC document is "presented as the current state of the art in requirements specification and evaluation" even though it "does not presume to include all possible security functional requirements but rather contains those that are known and agreed to be of value by the CC Part 2 authors at the time of release." The discussion here of flaws in the CC is not intended to suggest that the entire methodology be abandoned -- since extensions are permitted, a product can be enhanced with security technology beyond that found in the current CC incarnation. Such extensions would need to be formulated and applied where it is possible to do so. Only for aspects where extensions can not be provided shall it be deemed that those related security objectives cannot presently be assured. Other assurances within the CC may not be perfect, but they are still better than providing no assurances at all.

5.1 Common Criteria Evaluation Process

The CC evaluation process is described as a hierarchical paradigm where the Target of Evaluation (TOE) is identified as a "monolithic product containing hardware, firmware, and software." Or alternatively as "a distributed product that contains multiple separated parts...connected...through an internal communication channel." "TOE evaluation is concerned primarily with ensuring that a defined TOE Security Policy (TSP) is enforced over the TOE resources." "The TSP is, in turn, made up of multiple Security Function Policies (SFPs). Each SFP has a scope of control, that defines the subjects, objects, and operations controlled...the SFP is implemented by a Security Function (SF), whose mechanisms enforce the policy and provide necessary capabilities." The portions of the TOE that enforce the TSP are referred to as the TOE Security Functions (TSF). "The primary goal of the TSF is the complete and correct enforcement of the TSP over the resources and information that the TOE controls." The reference monitor concept (discussed in Section 3.6.1) and its embodiment as a reference validation mechanism are used to enforce the access control policies of a TOE. For the distributed product TOEs, the communication channels as well as remote products that are interfaced may need to be evaluated. Additional components of the TOE that should be considered include the set of interactions defined as the TSF Scope of Control (TSC) and the set of security function interfaces known as the TSF Interface (TSFI).

TOE resources can be structured into entities, each with specific security properties. Entities can be *active*, causing internal operations to be performed on information within the TOE, or *passive*, as the container that sends or receives information. Active entities are known as *subjects*, which can act on behalf of a single authorized user, or *processes* in behalf of multiple users, or as a *trusted process* of the TOE. Passive entities are known as *objects*, the targets of operations performed by subjects. A subject can also act as an object on occasion. Users, subjects, information and objects possess *attributes* that can be used to provide general information or specific access control content (security attributes) for use in enforcement of the TSP. Access control and information flow control SFPs base their policy decisions on these attributes. Additional data that has no special meaning for the TSF, but which is used in accordance with the TSP, is known as *user data*.

"The CC presents the framework in which an evaluation can take place." It establishes certain common constructs and the jargon to be used to express them. Constructs are organized into a class-family-component hierarchy. A *class* is "the most general grouping of security requirements" that share a common focus, though their security objectives may differ. A *family* is a group of "requirements that share security objectives but may differ in emphasis or rigour." A *component* "is the lowest level expression of security requirements, and is the indivisible security requirement that can be verified by the evaluation." Dependencies may exist between components, and individual components can be tailored for use through iteration (repetition), assignment of parameter values, selection of items, and refinement through additional detail.

Security requirement components can be grouped into packages. The EALs "are predefined assurance packages" containing "a baseline set of assurance requirements for evaluation." These are detailed in Part 3 of the CC. The Protection Profile is formulated from security requirements in the CC, or by explicit statements referencing the pertinent EAL. The PP "also contains the rationale for security objectives and security requirements." PPs are developed by users, developers, or other parties concerned with security requirements. "A PP gives consumers a means of referring to a specific set of security needs and facilitates future evaluation against those needs." The establishment of generalized, reusable PPs for voting system requirements, therefore, is viewed as an essential base for the development of consistent policies under which evaluations of proposed voting systems can be performed. This can be extended through the issuance of a Security Target document that "is the basis for agreement between all parties as to what security the [particular] TOE offers." This document would list the security features that have been designed into the voting system under scrutiny.

For example, at the lowest level (EAL1), functional testing is applied to systems "where some confidence in correct operation is required" and "due care has been exercised with respect to the protection of personal or similar information." According to the CC, "an evaluation at this level should provide evidence that the TOE functions in a manner consistent with its documentation, and that it provides useful protection against identified threats." This specification actually far exceeds the level of examination that is currently typical for many election installations. The CC assurance components identified for EAL1 are: version numbers; installation, generation and startup procedures; informal functional specification; informal correspondence demonstration; administrator guidance; user guidance; and independent conformance testing. Section 5.4 describes why such modest assurances are inadequate for certifying compliance with certain of the Shamos constraints, since this assurance level does not protect sufficiently against many potential threats. Examination of some of the higher EAL levels is similarly approached.

5.2 Voting System Criteria

At present, some subset of the misuse vulnerabilities described in Section 4.2 exists in all voting systems. Eradicating these, even if it were possible to do so, provides a solution to only part of the problem of providing a secure and accurate vote count. Neumann provides a set of generic criteria that must be assured for voting systems.[NEU95] This list does not track well with the CC, so it has been adapted into a framework of topics for this discussion. The following subsections expand on these topics through reference to (and quotations directly from) the relevant portions of the CC documentation. (Although this section is somewhat tedious, it represents a significant reduction from the original 700-page CC specification document, providing clarity for the assurance process through extraction of its salient components and association with voting applications where relevant. Such a reduction was non-trivial, and no other prior summary, either generic or specific to any particular secure application area, appears to currently exist. The entire CC specification was reduced to one-tenth of its original size by this author, but only the features relevant to voting systems are included here.) Where 'TOE' or 'system' appear in these descriptions, one should consider this as referring to the voting system under examination or as deployed for actual use. The focus in this section is on those aspects of voting systems that require security assurance. The remainder of this chapter revisits these criteria in terms of the Evaluation Assurance Levels, in conjunction with the constraints applied by Shamos' fundamental requirements. The following chapter deals with non-security issues involving balloting and vote tabulation.

A. System Requirements

The establishment of the security environment for the system takes into account: the operating environment, identifying all physical aspects relevant to security (which may include personnel arrangements); the protected assets to which security will be applied (data files as well as authorization credentials, and the voting implementation itself); and a description of the intended usage. Specific statements regarding the security policies, threats and risks would include: assumptions met by the environment in order for it to be deemed secure; identification of all relevant threats to security; and statements of relevant organizational security policies and rules. With regard to security threats: "the CC characterizes a threat in terms of a threat agent, a presumed attack method, any vulnerabilities that are the foundation for the attack, and identification of the asset under attack. An assessment of risks to voting security would qualify each threat with an assessment of the likelihood of such a threat developing into an actual attack, the likelihood of such an attack proving successful, and the consequences of any damage that may result."

B. Functionality

Functionality pertains to the proper operation of the voting system for its designated purpose(s). The CC addresses the overall functional aspects related to system security, but additional functional requirements should be applied and assessed for any operations related to ballot collection and vote tabulation. Within a CC evaluation, the summary specification requirement involves the production of "a high-level definition of the security functions claimed to meet the functional requirements and of the assurance measures taken to meet the assurance requirements." The developer provides a summary specification and rationale. This describes the security functions and the assurance measures, tracing back to the functional requirements so it can be seen which functions satisfy which functional requirements, that every security function contributes to at least one functional requirement, and that all functions work together in the satisfaction of these requirements. The same is done with the assurance measures and assurance requirements.

C. Correctness (Accuracy)

Correctness is a further refinement of functionality; whereas a voting system might have the appearance of being operational, it may not be functioning correctly at all, or at all times. Correctness of the security measures is assured through application of the CC evaluation. Correctness of other aspects is assured through proper use of design techniques, configuration management, testing, and addition of features such as fault tolerance and other ongoing correctness checks. This is particularly critical since some remedies may only be available for failures detected during the voting session.

The CC describes an underlying abstract machine test that "defines the requirements for the TSF's testing of security assumptions made about the underlying abstract machine upon which the TSF relies. This 'abstract' machine could be a hardware/firmware platform, or it could be some known and assessed hardware/software combination acting as a virtual machine." Some forms of tests of the abstract machine could include power-on for hardware/firmware and software elements, and loadable tests (such as processor component stress tests for logic, calculation and memory units). Periodic invocation of functions should be scheduled to test the security assumptions of the underlying abstract machine in off-line, on-line, or maintenance modes. Access to these functions may be limited to authorized users. Noted is that "this need for confidence that the TOE is operating correctly must be balanced with the potential impact on the availability of the TOE, as often times, self tests may delay the normal operation of a TOE."

D. Accountability

The CC shifts responsibility for assessment, and hence accountability, onto the purchaser, by saying "the owners of the assets will analyse the possible threats to determine which ones apply to their environment. The results are known as risks. This analysis can aid in the selection of countermeasures to counter the risks and reduce it to an acceptable level." "Countermeasures are imposed to reduce vulnerabilities and to meet security policies of the owners of the assets (either directly or indirectly by providing direction to other parties)." "Owners will need to be confident that the countermeasures are adequate to counter the threats to assets before they will allow exposure of their assets to the specified threats. Owners may not themselves possess the capability to judge all aspects of the countermeasures, and may therefore seek evaluation of the countermeasures." Such evaluation would result in an assigned assurance rating for the proposed countermeasures, which can be used to determine their effectiveness against the identified risks. Section 4.2 on voting system misuse in this thesis could certainly be used as a guide in developing a set of countermeasures, within feasibility, for the risks as identified therein.

E. Disclosability

The CC evaluation process is intended to result in "a confirmation that the Target of Evaluation satisfies its security requirements as stated in the Security Target." This confirmation takes the form of one or more reports that are issued to the developer and which should be made available to the voting system purchasers. "The degree of confidence gained through an evaluation depends on the assurance requirements (e.g., Evaluation Assurance Level) met."

The developer provides a statement of security requirements, along with the security requirements rationale. This statement identifies and justifies the security functional requirements and the security assurance requirements and, on the basis of this, selects a CC EAL for the product. Any security requirements for the IT environment are also identified. All completed and any uncompleted operations on security requirements are identified, along with dependencies among the requirements and discussion on why any non-satisfaction of dependencies should be allowed. There is a statement of the minimum strength of function level, and identification of any requirements for which an explicit strength of function is needed along with the specific metric. The security requirements rational demonstrates: that the minimum strength of function level plus any explicit strength of function claim is consistent with the security objectives; that the security requirements are suitable to meet the security objectives; and that the set of security requirements forms a mutually supportive and internally consistent whole. The evaluator confirms that the information meets all requirements for content and presentation of evidence, and that it is coherent and internally consistent.

F. Reliability

System reliability is of a global nature, and is addressed by the specific criteria on integrity, availability, fault tolerance, data requirements, testing, paths, and recovery, all covered below.

G. Integrity

Integrity involves showing that the identified assets of the voting system are secure and that security concerns are addressed "at all levels from the most abstract to the final IT implementation in its operational environment." The application of security concepts should be an ongoing and integral part of system development, not merely a set of patches to be applied to an existing product.

The CC addresses protection of the security functions as "requirements that relate to the integrity and management of the mechanisms that provide the TSF (independent of TSP-specifics) and to the integrity of TSF data (independent of the specific contents of TSP data)." The security functions are viewed as consisting of three portions: the abstract machine, which is a physical or virtual machine upon which the specific implementation executes; the implementation that executes the mechanisms enforcing the security policies; and the data that is contained in administrative databases and guides the enforcement of the security policies.

The CC's concerns with integrity include: the ability to detect external attacks on security-relevant parts of the system; verification of the correct operation of the underlying abstract machine, its data and executable code; implementation of a reference monitor for ensuring that the security features can not be bypassed; addressing behavior of the secured system when failure occurs and immediately after; protecting transmitted data; enabling replay of various types of information and/or operation; synchronizing states between different parts of the system if it is distributed; ensuring reliable timing; maintaining consistency of shared data; detecting and preventing unauthorized modification; using configuration management to track changes and detect integrity errors; acceptance and self-testing.

Configuration Management (CM) plays a strong role in ensuring "the integrity of the TOE from the early design stages through all subsequent maintenance efforts."

Objectives include: "ensuring that the TOE is correct and complete before it is sent to the consumer; ensuring that no configuration items are missed during evaluation; preventing unauthorised modification, addition, or deletion of TOE configuration items." Evidence of all of the above should be shown for voting products. A configuration list, uniquely identifying all configuration items maintained by the CM system should be provided, and evidence should be shown that the CM system tracks all listed items. "The evidence shall demonstrate that the CM system operates in accordance with the CM plan" via documentation, or a detailed demonstration by the developer. The responsibility for determining that this evidence is sufficient falls on the evaluator. A further requirement is "that the CM system support the generation of the TOE" by providing "information and/or electronic means to assist in determining that the correct configuration items are used in generating the TOE." There should be "no ambiguity in terms of which instance of the TOE is being evaluated." Acceptance procedures should be used "to confirm that any creation or modification of configuration items is authorised" and it "shall describe the procedures used to accept modified or newly created configuration items as part of the TOE." "The CM documentation shall include a configuration list, a CM plan, an acceptance plan, and integration procedures."

"Integration procedures help to ensure that generation of the TOE from a managed set of configuration items is correctly performed in an authorised manner." "Requiring that the CM system be able to identify the master copy of the material used to generate the TOE helps to ensure that the integrity of this material is preserved by the appropriate technical, physical and procedural safeguards." "The integration procedures shall describe how the CM system is applied in the TOE manufacturing process. The CM system shall require that the person responsible for accepting a configuration item into CM is not the person who developed it. The CM system shall clearly identify the configuration items that comprise the TSF. The CM system shall support the audit of all modifications to the TOE including as a minimum the originator, date, and time in the audit trail. The CM documentation shall demonstrate that the use of the integration procedures ensures that the generation of the TOE is correctly performed in an authorised manner. The CM documentation shall justify that the acceptance procedures provide for an adequate and appropriate review of changes to all configuration items." These integration procedure items, particularly that one specifying that the individual accepting an item into CM is not the person who developed it, are especially helpful in reducing the possibility of bogus code being introduced into the voting system just prior to release.

Other aspects of integrity assurance involve the use of audit trails and controls on accessibility, topics discussed earlier in this thesis (Sections 3.3 and 3.6 respectively).

H. Availability

Availability is addressed by the resource utilization section of the CC, which "provides three families that support the availability of required resources such as processing capability and/or storage capacity. The family Fault Tolerance provides protection against unavailability of capabilities caused by failure of the TOE. The family Priority of Service ensures that the resources will be allocated to the more important or time-critical tasks and cannot be monopolized by lower priority tasks. The family Resource Allocation provides limits on the use of available resources, therefore preventing users from monopolizing the resources." Fault tolerance is further discussed below. Since there is a timely nature of elections, availability for the duration of the voting session as well as the tabulation period, is critical.

I. Fault Tolerance

Fault tolerance ensures "that the TOE will maintain correct operation even in the event of failures." Degraded fault tolerance ensures the operation of certain capabilities when specific failures occur. Limited fault tolerance ensures operation of all capabilities when specific failures occur. A voting system must ensure that certain capabilities (such as ballot data retention) remain available in the event of failures (like power failure, hardware failure, or software error). The voting system must be capable of remaining in a secure state after a failure so that its relevant security policies continue to be enforced. Fault tolerance mechanisms can be active, invoking specific functions on error conditions, or passive, when the system architecture can handle the error (such as a multiple processor system where operation among the remaining processors can continue even in the absence of one). Failure can be accidental or intentional, but to the fault tolerant system this should not matter. The list of capabilities maintained by a degraded system after a fault is detected should be specified. Categories and types of failures that require resistance should be listed. Detected failures and discontinuation of capabilities should be audited.

Fail secure "ensures that the TOE will not violate its TSP in the event of certain types of failures in the TSF." A secure state is one "in which the TSF data are consistent and the TSF continues correct enforcement of the TSP" -- this should be defined by the developer in the security policy model. Auditing all situations in which failure with preservation of secure state occurs may not be possible, so the designer should specify which situations are feasible and desirable for auditing. Hard failures involving equipment malfunction and subsequent maintenance service or repair, as well as certain soft failures that require initialization or resetting should be listed if they must be fail secure ("should preserve a secure state and continue to correctly enforce the TSP").

J. Data Requirements

Information is regarded by the CC as assets, some of which must be strictly controlled. Such control may also be applied to data dissemination and modification. Data collection, tabulation, storage and maintenance are some aspects where controls would be necessary in a voting system. An extensive discussion of applicable data requirements appears in Section 3.6.2. Other data issues involving confidentiality, retention and recountability are addressed in this section, directly below. The CC discusses security countermeasures that may involve layers of properly operating IT security products. The IT system accreditor is charged by the information owner to "determine whether the combination of IT and non-IT security countermeasures furnishes adequate protection of the data, and thus to decide whether to permit the operation of the system."

K. Confidentiality

Confidentiality issues necessitate that a delicate balance be maintained between requirements for ballot privacy along with voter authentication assurance. This area is essential to the voting application, particularly if on-line identification of voters is involved. Although the CC permits secure systems to occasionally violate privacy, this should not be allowed in certain parts of the voting application. Four aspects, whose interrelationship may be complex, are described: anonymity, pseudonymity, unlinkability, and unobservability. Underlying conflicts with these aspects and other security requirements are discussed here, as well as later in this chapter. The system must provide protection from inadvertent disclosure of confidential data, and against individual users and subjects acting alone or cooperating to discover private information. There is an implicit assumption that users will not deliberately expose confidential data, creating an inherent vulnerability that is not addressed by the CC. This vulnerability could be exploited in vote-selling or other subversions.

Anonymity "ensures that a user may use a resource or service without disclosing the user's identity. The requirements for anonymity provide protection of the user identity." It is critical to note that anonymity is not intended to protect the identity of the processes working on behalf of the user, only "that other users or subjects are unable to determine the identity of a user bound to a subject or operation." The system "must ensure that a specified set of users and/or subjects are unable to determine the real user name bound to a list of subjects and/or operations and/or objects." In the more stringent case, the system shall also provide certain services to certain subjects without soliciting any reference to the real user identity. The CC allows that certain authorised users may be excluded from this restriction, and hence can retrieve a user's identity, but this may be improper in a voting setting. Systems may provide anonymity for all or just some subjects/operations. Notes in the CC on anonymity include the statement that "possible applications include the ability to make enquiries of a confidential nature to public databases, *respond to electronic polls*, or make anonymous payments or donations." (Italics added by this author.) In simple anonymity, the set of users and/or subjects against which the system must provide protection should be specified. In anonymity without soliciting information, the list of services subject to the anonymity requirement should be identified, along with "the list of subjects from which the real user name of the subject should be protected when the specified services are provided."

Pseudonymity "ensures that a user may use a resource or service without disclosing its user identity, but can still be accountable for that use." This accountability can be through a reference (alias) held by the TSF, or by a provided alias such as an account number. Pseudonymity is similar to anonymity, but the difference is that here the "reference to the user's identity is maintained for accountability or other purposes." "Pseudonymity requires that a set of users and/or subjects are unable to determine the identity of a user bound to a subject or operation, but that this user is still accountable for its actions." Like anonymity, pseudonymity involves the system ensuring that certain users or subjects are unable to determine the real user name bound to certain subjects, operations, and/or objects. Auditing of requests of resolution of user identity may be necessary. Variations of pseudonymity are possible, such as reversible pseudonymity, which requires the system to determine the original user identity, under certain conditions, based on a provided alias. The reference can also be used by other users for various purposes (such as collecting statistical data) without obtaining the aliased user's identity. Appropriate application areas include electronic cash systems, where billing would occur to the alias, but the use of the system could remain anonymous. For reversible pseudonymity, the users, subjects, and conditions that can reveal the true user name should be specified. (Noted here is the example administrative condition "such as on a court order.") In another variation, alias pseudonymity, the system determines and accepts aliases for users and verifies conformance to the specified alias metric. The system may provide an alias that is identical to one used earlier, under certain conditions, or otherwise the alias provided is unrelated to previous aliases. The set of users against which the system must provide protection, the list of subjects, objects and/or operations where the real user name is protected, and the types and conditions placed on aliases used, provided or generated by the system, should be specified.

Reversible pseudonymity is often suggested as an auditing mechanism by electronic voting system vendors. The scheme described typically involves the issuance of a number (PIN) which can be used by the voter after the election to "look up" their own ballot to be sure it was recorded correctly. This is problematic for a number of reasons. First, if one can view their ballot, then they could possibly sell their vote -- it need not be the case that a specific voter name be identified along with the ballot -- holding the PIN is proof enough. Secondly, if a voter determines that their ballot was recorded incorrectly, what recourse do they have for correction? And third, looking up any individual ballot or set of ballots in no way ensures that these ballots were tabulated properly -- this would only be possible to ascertain if everyone voluntarily revealed their choices, thus violating the concept of an anonymous election.

Unlinkability "ensures that a user may make multiple uses of resources or services without others being able to link these uses together." This is especially pertinent to anonymous voting in a setting that also includes authorization capabilities. "Unlinkability requires that users and/or subjects are unable to determine whether the same user caused certain specific operations in the system." In particular, a certain set of users and/or subjects may be required to be unable to determine whether certain operations were caused by the same user, under certain relationships. "Unlinkability differs from pseudonymity [in] that, although in pseudonymity the user is also not known, relations between different actions can be provided." The concept is that profiling of operation usage, or relationships between different operation invocations by the same user, should be prevented by protecting the user identity. "Hiding the relationship between different invocations of a service or access of a resource will prevent this kind of information gathering." The types of protected relationships will be specified so that usage patterns that disclose a user's identity can be prevented from being formed. The example given is a series of anonymous phone calls made by the same user. The set of users, the list of operations, and the relationships among operations and among identities that should be protected should be specified.

For voting, it should be the case that the voter authorization components be unlinkable to the ballot casting sections. This too is problematic. Even if authorization invokes an unobservable transfer to a wholly disjoint balloting system, a denial of service interruption could prevent completion of ballot casting, disenfranchising the voter if they were marked as having been permitted access to vote. Communication back to the authorization system of ballot casting completion by the voter would involve retention of identity, which voids unlinkability.

Unobservability "ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource or service is being used."

Other components of unobservability may include requirements that: "users and/or subjects can not determine whether an operation is being performed"; there be provisions for "specific mechanisms to avoid the concentration of privacy related information"; the system should "not try to obtain privacy related information that might be used to compromise unobservability; and certain users may have "a capability to observe the usage of resources and/or services." (This observation capability should not be used for secret balloting, of course.) Unobservability may need to be maintained for certain users and/or subjects doing certain operations, throughout or through some portion of the lifetime of the data. Unobservability differs from the other three privacy types because "the intent is to hide the use of a resource or service, rather than to hide the user's identity."

Techniques suggested for implementation of unobservability include: distribution of information impacting unobservability; broadcast of information widely so that users cannot determine who actually received and used it (a rather peculiar solution); and cryptographic protection along with message padding. Authorized users may need to see that certain resources are being used, although other users may be protected from viewing the use. Specification of the list of users and/or subjects, the list of operations, the list of objects, and the set of protected users and/or subjects should be made. The system itself may be prohibited from soliciting information that might be used to compromise unobservability. Areas of controlled privacy information and the conditions and users that can allow for its dissemination should also be specified. An unobservability requirement may also necessitate covert channel analysis.

The CC notes that: "Unobservability of communications may be an important factor in many areas, such as the enforcement of constitutional rights, organizational policies, or in defense related applications." This is one of only a few places in the CC where application usage is suggested, and in this case, is specifically related to the voting problem. The CC even describes a voting scenario in its discussion of unobservability thusly: "A more complex example can be found in some 'voting algorithms'. Several parts of the TOE will be involved in the service, but no individual part of the TOE will be able to violate the policy. So a person may cast a vote (or not) without the TOE being able to determine whether a vote has been cast and what the vote happened to be (unless the vote was unanimous)." Although in theory this seems plausible, the entire concept is fatally flawed. If the TOE can not determine whether a vote has been cast or what thecontents are, how can it be possible to assure ballot data correctness? This matter is unresolved.

L. Retention and Recountability

Data retention is an important component of the voting system. Although voter authentication and authorization is currently performed through a manual process (for absentees as well as on-site voters), computerization of this aspect is separately possible (although it creates additional risks). If automatic authorization is provided, voters who have cast ballots need to be flagged if they attempt to re-access the system in the same voting session, hence these records must be kept current and accessible. (Automatic authorization is of course subject to the issues discussed in the prior section on confidentiality.) If the voter information is distributed, there must be some real-time reconciliation process in order to maintain consistency.

Recounts should be able to be performed independent of the voting system used for ballot entry. The availability of different recount systems (provided perhaps by each political party, as well as other interested agencies, such as the League of Women Voters and news bureau services) introduces additional "checks and balances." These recount systems could be created separately (possibly via open source code releases) and pre-certified (before the election) for use by demonstrating correctness on test data sets. (Granted, this form of black-box testing is not comprehensive, but it would be hoped that since multiple systems were operating on the same ballot data, discrepancies in operations would be revealed.) There would need to be some procedure for resolving discrepancies if different election results were obtained from the diverse systems.

The entire set of voter and ballot related data for the voting session should be retained possibly indefinitely, or at least until the time when recounts can not be requested in litigation or other disputes. Permanent and non-volatile media should be used for electronic data retention, with redundancy (preferably on separate media and in the form of recoverable checksums) provided as necessary. Physical (paper) ballots should be maintained appropriately, in order to eliminate or reduce suspicions of tampering.

M. User Requirements

From the view of the CC, users are external to the system, and thus fall outside of the security functions' scope of control. The user interactions through the security function interfaces (whether via local or remotely situated humans, or external IT entities) are of interest to the security functional requirements. The period of interaction of a user with the security functions is known as a session. These interactions can be subject to controls, such as: "user authentication, time of day, method of accessing the TOE, and number of allowed concurrent sessions per user." An authorised user is one "who possesses the rights and/or privileges necessary to perform an operation." When there are certain administrative duties that need to be performed, the identification of roles, as "a pre-defined set of rules establishing the allowed interactions between a user and the TOE" may be necessary. Here, the users would include voters, as well as all personnel and agents that are allowed to interact with the system, prior to, during, and after an election.

N. Administrator Requirements

Administrative personnel integrity is assured through the assignment and application of security management roles, with the intention of reducing "the likelihood of damage resulting from users abusing their authority by taking actions outside their assigned functional responsibilities. It also addresses the threat that inadequate mechanisms may have been provided for secure administration. Information must be maintained to identify whether a user is authorized to use a particular security-relevant administrative function." The roles can encompass a set of capabilities or just a single right, which should be specified. Auditing of role assignments should occur. Some roles may be mutually exclusive, and two-person control can be specified. Explicit requests to assume certain specific roles may be required. The different roles that the system should recognize (such as entity owner, administrator, and other users) should be specified, along with conditions regarding how these roles would be managed, as well as interrelationships and restrictions (such as separation of capability) between different roles. The roles addressed here would be those specifically related to system security. The CC states that "the PP/ST author should specify the conditions that govern role assignment" but it does not elaborate on the particulars of dealing with users who have obtained multiple roles or the prevention of this from happening -- these issues appear to be left to the developer's discretion. Note that when a distributed system is involved, timely propagation of changes in attributes and role information may be problematic.

Security management is used to specify the control of security data, attributes, functions, and definition of roles. Security attributes may be established through appropriate default settings or assignment of values which are checked for validity within the secure state. There may be a requirement that values assigned to security attributes be such that the system will remain in a secure state (although here, the definition of 'secure' may depend upon which role's attributes are being assigned). The security attributes (such as user groups, roles, rights, process priorities, and so on) might be assigned to be managed by the user themselves, a subject, or a specific authorised user. "It is noted that the right to assign rights to users is itself a security attribute and/or potentially subject to management." Only certain authorized users may be permitted access to security data. Modifications should be subject to audit.

Certain users may be able to revoke authority of others. Management of revocation functions includes: the roles that can invoke revocation; the users, subjects, objects and other resources subject to revocation; and the rules for revocation. Auditing may include successful and unsuccessful attempts at revocation. Revocation may occur: on next login of the user; on the next attempt to open a file; or within a fixed time. Time limits for the validity of security attributes may also be enforced. An authorised user may specify the expiration times for specific attributes. The attributes that are subject to expiration, the roles that can modify these attributes, along with the actions taken when the expiration time has passed may be handled by management functions. Auditing may include tracking the expiration time specifications and actions taken due to expiration. Secure time stamps are essential for this function.

Function management is necessary to enable authorized users to set up and control secure operations. The categories of administrative functions may include: access control; accountability and authentication controls; auditing system controls; per-user policy attributes; system access controls; control and management of user groups; functions that relate to controls over availability; general installation and configuration; and routine control and maintenance of resources. Roles may be identified to manage the security functions, including specific roles allowed to obtain the current status, disable or enable security functions, or modify their behaviors.

Data management imposes requirements on certain information (like time stamp and audit trail) regarding access (read, delete, create) by certain user roles. The operations that can be applied to the identified data should be specified, along with the roles that can perform the operations. There is the concept of clearing data which involves removing the information itself while maintaining system operation. The definition of administrator functions involved with clearing data, especially in a repeated balloting scenario, may be an important issue.

All of the administrator requirements are critical in the voting setting because alterations in ballot data or tallies, or linking of voter authentication information to ballots cast, can have serious consequences. Administrative roles should be assigned to agents from the different political parties, who would monitor the entire process. Although collusion is still possible, here again, multi-partisan checking can provide a way to reduce opportunities for such occurrences.

O. Interface Usability

Interfaces include those within the secure voting system, those between components of a distributed system which may involve both secure and insecure products, and those between the users (voters and administrators) and the system. The interfaces that relate to interactions involving security functions should be described and detailed. The CC assurance levels (see Section 5.3) address basic assurance through interface specification and higher levels of assurance through complete interface analysis. Data interface issues, as well as others mentioned above under integrity (such as timing, synchronization, and consistency) must be addressed. The user interface must also be examined in terms of clarity, correctness, friendliness, accessibility (for disabled as well as traditional users), as well as security. Since the ballot interface could overtly or inadvertently influence voter choices, its design should be done in accordance with well-established election guidelines. Ballot layout should go through a multi-partisan approval process prior to each election.

P. Documentation

Guidance documents are necessary for secure administration and use of the system. The CC describes numerous relevant aspects for such documentation.

Administrator guidance "refers to written material that is intended to be used by those persons responsible for configuring, maintaining, and administering the TOE in a correct manner for maximum security." Persons responsible for performing these functions are necessarily trusted. These guidance documents help the administrators "understand the security functions provided by the TOE, including both those functions that require the administrator to perform security-critical actions and those functions that provide security-critical information." Warnings to users with regard to the security environment and the security objectives should be appropriately covered in the administrator guidance. Where an administrator has control over security parameters, guidance needs to be provided on secure and insecure settings. The administrator guidance should include descriptions of: the administrative functions and interfaces; how to administer the system in a secure manner; warnings about functions and privileges that should be controlled; all assumptions regarding user behavior that are relevant to secure system operation; all security parameters and secure values as appropriate; each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics under system control; and security requirements for the IT environment that are relevant to the administrator. The guidance documents should be consistent with all other documentation supplied for assurance evaluation.

User guidance "refers to material that is intended to be used by non-administrative human users of the TOE, and by others (e.g., programmers) using the TOE's external interfaces. User guidance describes the security functions provided by the TSF and provides instructions and guidelines, including warnings, for its secure use. User guidance provides a basis for assumptions about the use of the TOE and a measure of confidence that non-malicious users, application providers and others exercising the external interfaces of the TOE will understand the secure operation of the TOE and will use it as intended." As in administrator guidance (above), any warnings to the users about the security environment and objectives should be appropriately covered. This may be "provided in separate documents: one for human users, and one for application programmers and/or hardware designers using software or hardware interfaces." (Note that it is the CC, and not this author, making the distinction between humans and programmers!) The user guidance shall describe: the functions and interfaces available to non-administrative users; the use of user-accessible security functions provided by the system; warnings about user-accessible functions and privileges that should be controlled; all user responsibilities necessary for secure operation, including assumptions regarding user behavior; and all security requirements for the IT environment that are relevant to the user. This guidance shall also be consistent with other documentation supplied for assurance evaluation.

Municipalities may require that some subset of the user guidance documentation be provided to voters, so these documents need to provide specific directions in a highly user-friendly fashion. Although the CC does not explicitly address training of users and administrators, such should be provided to supplement the documentation where appropriate or necessary, especially for administrators who are unfamiliar with secure computer system operations, and users who may feel uncomfortable with the new equipment.

Q. Testing

Testing also applies to that which is performed by the developer during final construction, and to product acceptance testing by independent examiners prior to initial deployment. Different types of tests are required by the various EALs for assurance at each level. Such tests should be constructed to follow standard procedures and practices for secure systems. The CC also addresses security function self test, which "defines the requirements for the self-testing of the TSF with respect to some expected correct operation." Tests may be performed at start-up, periodically, on request of an authorized user, or when other conditions are met. Other families define the actions taken by the system as the result of self-testing. These tests are used to detect corruption of security function executable code and data by unforeseen failure modes, design oversights, or malicious corruption. Use of security function self-test may help prevent unwanted changes to an operational system during maintenance activities. The tests may only be accessible to authorized users in off-line or maintenance modes. Self-testing must be enabled throughout the voting session, with immediate fail-safe alert and disabling of voter access when any operational or security defect is noted. Self-monitoring always provides the risk of failure to report during equipment breakdown (if the equipment is malfunctioning, so also may be the self-monitoring components), so independent checking is also necessary although it may be difficult to implement.

R. Paths

Trusted paths and trusted channels provide an approach for dealing with secure communication between the security functions and remote IT products. (Voting issues related to trust and secure channels were discussed in Sections 3.6.1 and 3.7 respectively.) Confidence is established that a user is communicating directly with the security functions, and that untrusted applications cannot intercept or modify the user's response. "Absence of a trusted path may allow breaches of accountability or access control in environments where untrusted applications are used. In voting, the absence of trusted paths enables data rerouting or spoofing possibilities." Private information, such as passwords, could be intercepted and used to gain access to the system. "As a consequence, responsibility for any system actions cannot be reliably assigned to an accountable entity. Also, these applications could output erroneous information on an unsuspecting user's display, resulting in subsequent user actions that may be erroneous and may lead to a security breach."

Rules can be defined "for the creation of a trusted channel connection that goes between the TSF and another trusted IT product for the performance of security critical operations between the products." It should be specified "whether the local TSF, the remote trusted IT product, or both shall have the capability to initiate the trusted channel," and also the functions that require a trusted channel should be specified.

The "requirements to establish and maintain trusted communication to or from users and the TSF" when required for security-relevant interaction should be defined, "either for initial authentication purposes only or for additional specified user operations." The "trusted path exchanges may be initiated by a user during an interaction with the TSF, or the TSF may establish communication with the user via a trusted path." Specification should be made regarding: whether the trusted path must be extended to remote and/or local users; whether the security functions, local and/or remote users should be able to initiate the trusted path; and whether the trusted path is used for initial user authentication and/or other particular services as identified.

S. Facility Management

Facility management may include physical plant maintenance (such as climate-control monitoring) as well as ensuring that supplies are available and operators are on call as necessary. The security aspects of physical protection are addressed by the CC when it refers to "restrictions on unauthorised physical access to the TSF and to the deterrence of, and resistance to, unauthorised physical modification, or substitution of the TSF." These controls are intended to "ensure that the TSF is protected from physical tampering and interference." Components may be included in order to passively and unambiguously detect or actively resist such tampering, along with required responses by the security functions in the case of automatic notification of tampering attempts. The capability to determine whether physical tampering with the security devices or elements has occurred should be provided via monitoring. The user role that gets informed in the case of intrusion, the devices that perform the notification, and the automatic responses to tampering, may all require management. Auditing of intrusion detection should be performed. Additionally, all roles which perform facility management tasks should be audited when system use is involved.

With voting, facility management is necessary during the time before an election (when equipment and materials are being prepared), during the election period (while ballots are being cast), and afterwards (when tallying, recount and contesting activities occur).

There is an additional concern that equipment remain secure during the long periods of time between elections, when opportunities for large-scale tampering could occur.

T. Recovery

Trusted recovery ensures "that the TSF can determine that the TOE is started up without protection compromise and can recover without protection compromise after discontinuity of operations. This is important because the start-up state of the TSF determines the protection of subsequent states." Various forms of recovery are possible: manual (human intervention is involved in order to return to a secure state), automated (secure state can be established without human intervention), automated recovery without undue loss (undue loss of protected objects is disallowed), function recovery (successful completion or rollback of data to a secure state is ensured). Each of these types of recovery may require different handling for resumption to occur, including in some cases entering into a maintenance mode where a secure state can then be resumed. Management is needed for personnel who can access the restore capability, as well as the list of failures that will be handled automatically. Auditing involves detection of failures of security functions and their types, along with tracking of resumption or failure thereof. System failure and data losses can result in disenfranchisement of voters who do not realize their ballot was not retained for tabulation (and who may not be able to be notified because the lost ballot is anonymous), or who may not have time to wait for the system to be returned to operation.

The recovery functions are therefore critical, and they should also be fast in order to minimize downtime.

Rollback "involves undoing the last operation or a series of operations bounded by some limit, such as a period of time, and return to a previous known state. Rollback provides the ability to undo the effects of an operation or series of operations to preserve the integrity of the user data." In the voting scenario, once a ballot is cast, no rollback should be permitted that could void or eliminate the ballot. During a balloting session, though, a voter might want to clear the ballot face of some or all selections in order to start over, prior to final ballot casting, so rollback of these operations should be permitted. In the manual lever machine scenario, during primary elections, a common problem that occurs is the accidental mis-selection of the political party by the poll workers in setup of the machine for an individual voter. This is generally resolved by waiting for another voter (of the party erroneously selected) to go ahead with that machine setting, and then the machine is properly set for the voter who was to have entered the booth. Depending on how selection of ballots is made for parties in an electronic setting, such rollback might be helpful here as well. In any event, rollback operations should be minimized for the voting application, as they provide avenues for covert operations, as well as reduced auditing capabilities.

U. System Distribution

The delivery and operation section of the CC "provides requirements for correct delivery, installation, generation, and start-up of the TOE." As it is essential that the voting systems in use are identical to those that were certified for purchase, this section must not be overlooked.

Delivery requires "system control and distribution facilities and procedures that provide assurance that the recipient receives the TOE that the sender intended to send, without any modifications. For a valid delivery, what is received must correspond precisely to the TOE master copy, thus avoiding any tampering with the actual version, or substitution of a false version." Delivery documentation should describe all procedures necessary to maintain security when distributing versions of the system to a user's site. The documentation should also describe how modifications or discrepancies will be detected, and further will describe how attempts to masquerade as the developer will be detected. A stricter version would also prevent as well as detect modifications.

The installation, generation and start-up documentation and procedures help ensure that the system has been installed, generated, and started up in a secure manner as intended by the developer. This requires a secure transition from the development system under configuration control to its initial operation in the user environment. Application of these requirements may differ depending on whether the system is delivered as operational, or brought up at the owner's site. Activities may take place at one location or may be distributed, depending on the type of setup required. Generation is applicable only if the system has the ability to generate portions of the operational instance from its implementation representation. If a generation log is required, the documentation should describe procedures for its creation, along with the manner in which it can be determined from the log how and when the system was generated. The installation, generation and start-up activities may be embodied in separate documents or grouped with other administrative procedures.

V. Compliance with Laws and Regulations

Assurance would be provided through examination of the system design documentation and other relevant descriptions with a list of regulatory requirements, which should be supplied by the purchaser at the time of placement of the procurement agreement, or even during the bidding process. Iteration on these points should occur early in the design process in order to fend off possible difficulties, and to make all parties aware of necessary constraints. Verification that compliance has occurred would be performed by the Independent Examiners during acceptance testing. Ambiguities in laws and regulations pertaining to voting systems need to be resolved so that clear compliance can be assured.

5.3 Evaluation Assurance Levels

The general intent of each of the seven Common Criteria Evaluation Assurance Levels is summarized in this section. The distinct components to be included in each level are provided via a set of tables in the CC documentation that also indicate the aspects relevant to voting systems. The seven EALs are hierarchically ordered, each representing more assurance than the lower numbered ones. As the CC says, "the increase in assurance from EAL to EAL is accomplished by substitution of a hierarchically higher assurance component from the same assurance family (i.e., increasing rigour, scope, and/or depth) and from the addition of assurance components from other assurance families (i.e., adding new requirements)." Each EAL can have no more than one component of any assurance family and all dependencies must be addressed. Other combinations of assurance may be created by augmenting the requirements with higher ones from the same family, but one can not delete or reduce any components. An EAL can be further extended with the addition of other (non-CC) explicitly stated assurance requirements.

A summary of the EALs is presented here in order to assist the reader in understanding the discussion that follows.

EAL1, Evaluation Assurance Level 1 -- functionally tested

"EAL1 is applicable where some confidence in correct operation is required, but the threats to security are not viewed as serious. It will be of value where independent assurance is required to support the contention that due care has been exercised with respect to the protection of personal or similar information." This level of evaluation could be conducted without assistance from the voting system developer, and at minimal cost. "EAL1 provides a basic level of assurance by an analysis of the security functions using a functional and interface specification and guidance documentation to understand the security behaviour. The analysis is supported by independent testing of the TOE security functions. This EAL provides a meaningful increase in assurance over an unevaluated IT product or system."

EAL2, Evaluation Assurance Level 2 -- structurally tested

EAL2 is applicable where "developers or users require a low to moderate level of independently assured security in the absence of ready availability of the complete development record. Such a situation may arise when securing a legacy system, or where access to the developer may be limited." EAL2 provides assurance by adding to EAL1 an analysis of the high-level system design. The independent testing also includes "evidence of developer testing based on the functional specification, selective independent confirmation of the developer test results, strength of function analysis, and evidence of a developer search of obvious vulnerabilities." A configuration list for the system and evidence of secure delivery procedures are also required.

EAL3, Evaluation Assurance Level 3 -- methodically tested and checked

EAL3 allows the developer to gain "assurance from positive security engineering at the design stage without substantial alteration of existing sound development practices." It is applicable where a moderate level of independently assured security is required, and when the system and its development need to be thoroughly investigated without substantial re-engineering. Here, the developer testing is based on the functional specification and high-level design, and assurance is provided through the additional use of development environment controls and configuration management. It requires "more complete testing coverage of the security functions and mechanisms and/or procedures that provide some confidence that the TOE will not be tampered with during development."

EAL4, Evaluation Assurance Level 4 -- methodically designed, tested, and reviewed

EAL4 gains assurance from good commercial development practices without requiring substantial specialist knowledge, skills, and other resources. "EAL4 is the highest level at which it is likely to be economically feasible to retrofit to an existing product line." It is applicable where a moderate to high level of independently assured security is required, and additional expenditures are permitted. This assurance requires an analysis of the complete interface specification, the high and low level system design, and a subset of the implementation, in addition to the lower EAL analysis, in order to understand the security behavior. "Assurance is additionally gained through an informal model of the TOE security policy." In addition to the lower EAL independent testing, "an independent vulnerability analysis demonstrating resistance to penetration attackers with a low attack potential" is also required. Development environment controls including automated configuration management must be used. The increase in assurance is thus gained by "requiring more design description, a subset of the implementation, and improved mechanisms and/or procedures that provide confidence that the TOE will not be tampered with during development or delivery."

EAL5, Evaluation Assurance Level 5 -- semiformally designed and tested

At this level, assurance is gained from rigorous commercial development practices supported by moderate application of security engineering techniques. Systems at this level will likely have been designed with EAL5 requirements in mind, and hence the additional costs will not be large. It is applicable where a high level of independently assured security is required without incurring unreasonable costs. The analysis will include all of the implementation, in order to understand the security behavior. Additionally required is "a formal model of the TOE security policy and a semiformal presentation of the functional specification and high-level design and a semiformal demonstration of correspondence between them. A modular TOE design is also required." The vulnerability analysis should show resistance to penetration attackers with a moderate attack potential. "The analysis also includes validation of the developer's covert channel analysis." The development environment should include comprehensive configuration management. The increase in assurance at this level is obtained "by requiring semiformal design descriptions, the entire implementation, a more structured (and hence analysable) architecture, covert channel analysis, and improved mechanisms and/or procedures that provide confidence that the TOE will not be tampered with during development."

EAL6, Evaluation Assurance Level 6 -- semiformally verified design and tested

"EAL6 permits developers to gain high assurance from application of security engineering techniques to a rigorous development environment in order to produce a premium TOE for protecting high value assets against significant risks. EAL6 is therefore applicable to the development of security TOEs for application in high risk situations where the value of the protected assets justifies the additional costs." A structured presentation of the implementation, along with a semiformal presentation of the functional specification, high and low level design, and semiformal demonstration of correspondence between them, is required. The system design must be both modular and layered. Evidence of resistance to penetration attackers with a high attack potential should be provided, along with validation of the developer's systematic covert channel analysis. A structured development process must be used, along with complete automation of configuration management. The additional assurance is obtained "by requiring a more comprehensive analysis, a structured representation of the implementation, more architectural structure (e.g. layering), more comprehensive independent vulnerability analysis, systematic covert channel identification, and improved configuration management and development environment controls."

EAL7, Evaluation Assurance Level 7 -- formally verified design and tested

This level is applied to "extremely high risk situations and/or where the high value of the assets justifies the higher costs." It is "currently limited to TOEs with tightly focused security functionality that is amenable to extensive formal analysis." This level requires "a formal presentation of the functional specification and high-level design, a semiformal presentation of the low-level design, and formal and semiformal demonstration of correspondence between them, as appropriate." The system design must be modular, layered, and simple (although it should be noted that the meaning of 'simple' is not defined in the specification). Evidence of developer testing based on implementation representation along with complete independent confirmation of the developer test results is required in addition to the lower level analysis. The increase in assurance is obtained "by requiring more comprehensive analysis using formal representations and formal correspondence, and comprehensive testing."

The security assurance requirements detailed in Part 3 of the Common Criteria document include seven classes as follows: configuration management, delivery and operation, development, guidance documents, life cycle support, tests, and vulnerability assessment. Each of these classes contains specific assurance families, and of these families, various assurance components are provided. The lowest component in each family that must be included in assurance for each of the seven EAL levels is indicated, in the CC, via a table (CC Part 3 Table B.1 appears here as Table 3). This table indicates the minimum assurance component numbers required for each assurance family for the EALs. Cells left blank indicate that no component is required at that level. Assurance at a particular EAL level can be required to go beyond the minimal set of assurance families, but if we are to apply the EAL levels to voting systems as the CC intended, we should consider only those families that the Criteria mandates.

Using this table, it is clear that EAL4 is the lowest level that should be used to certify voting systems, since all lower levels omit salient requirements involving the development process. EAL4 does not include any covert channel analysis, which first appears in EAL5, so perhaps the higher level should be used as the minimal assurance evaluation standard. This would agree with the intent of the EALs, since EAL5 is noted as the lowest level that would have been targeted for newly designed systems (all lower levels are applicable to security assurance retrofits). EAL5 requires all assurance families, with the exception of flaw remediation (in life cycle support), which is for some unknown reason omitted from all seven EAL levels, as it turns out. Since the attack potential of voting systems is likely to be high, the more stringent EAL6 evidence of resistance should also be included.

Assurance Class	Assurance Family	EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Configuration Management	Automation				1	1	2	2
	Capabilities	1	2	3	4	4	5	5
	Scope			1	2	3	3	3
Delivery and Operation	Delivery		1	1	2	2	2	3
	Install, Gen., Start-up	1	1	1	1	1	1	1
Development	Functional Spec.	1	1	1	2	3	3	4
	High-Level Design		1	2	2	3	4	5
	Implementation Rep				1	2	3	3
	TSF Internals					1	2	3
	Low-Level Design				1	1	2	2
	Rep Correspondence	1	1	1	1	2	2	3
	Security Policy Mod				1	3	3	3
Guidance Documents	Administrator	1	1	1	1	1	1	1
	User	1	1	1	1	1	1	1
Life Cycle Support	Develop. Security			1	1	1	2	2
	Flaw Remediation							
	Life Cycle Defn.				1	2	2	3
	Tools & Techniques				1	2	3	3
Tests	Coverage		1	2	2	2	3	3
	Depth			1	1	2	2	3
	Functional		1	1	1	1	2	2
	Independent	1	2	2	2	2	2	3
Vulnerability Assessment	Covert Channel					1	2	2
	Misuse			1	2	2	3	3
	TOE Security Fun.		1	1	1	1	1	1
	Vulnerab. Analysis		1	1	2	3	4	4

Table 3 – Cross Reference of EALs and Assurance Components

What the Common Criteria document does not provide, in its discussion of EAL levels, is a mapping back to the security functional components of CC Part 2. Part 2 details eleven classes: security audit, communication, cryptographic support, user data protection, identification and authentication, security management, privacy, protection of the TSF, resource utilization, TOE access, and trusted path/channels. Each of these classes also contain families and leveled components of each family. Although CC Part 2 contains a table which indicates dependencies between families, the connection between Part 2 and Part 3 is weak. It should be assumed, then, that a developer would include all Part 2 classes and families required for their secure product, and that the Part 3 assurance analysis would reveal errors and omissions. In this thesis, the criteria described in Section 5.2 intentionally track the CC Part 2 classes and families. The analysis indicates that nearly all of the Part 2 families require inclusion in voting systems.

In summary then, the evaluation assurance methodology for new voting systems should use the EAL5 assessment with EAL6 attack resistance and, as a part of the design analysis, should assure that the Part 2 families, as addressed in Section 5.2 in this thesis, were all satisfied. Expansion from the CC would be necessary, as described, where implementation is an issue (such as with cryptographic algorithms, automated intrusion and misuse analysis, and so on). Alternatively, EAL4 assessment could be applied (though it is certainly weaker) to existing systems seeking CC certification, with additional covert channel and high attack resistance analysis.

5.4 Generic Security Assessment Questions

In light of the above discussion, certain generic questions emerge for the evaluation of secure products. These are not particular to the voting setting, but can be used as the basis of an assessment methodology for electronic vote tabulation systems, in conjunction with the CC EAL components. The list presented here can be augmented with additional items or finer detail, as necessary.

1. What are the assets that require security protection?
2. What security risks have been identified, and what is the likelihood of each?
3. What countermeasures have been specified to deal with the identified risks?
4. What security assurance level has been selected for the system? Justify the appropriateness of this rating. How has conformance been established?
5. What assumptions are made about the operating environment in order for it to be deemed secure?
6. What are the policies and rules required to enforce security?
7. What are the specified security functions and assurance measures? Have these been traced back to the functional requirements to ensure that coverage is comprehensive?

8. Has a security requirements rationale document been presented? Does it demonstrate consistency with the security objectives for the system? Is the rationale comprehensive and consistent? Are any objectives unsatisfied, and if so, why?

9. What are the integrity concerns, and how have these been addressed?

10. What procedures are in place for secure system development? How have these been enforced and documented?

11. What are the resource allocation, priority of service, and fault tolerance policies and procedures?

12. What are the data requirements, and how are these implemented and enforced?

13. What are the data retention policies and procedures?

14. Have all communication paths been identified and secured as appropriate?

15. What are the confidentiality requirements, and how are these implemented and enforced?

16. What are the user roles? How are rules applied and enforced with the roles?

17. What are the authentication, authorization, and access control policies? How are these applied and enforced?

18. What are the administrative tasks and responsibilities?

19. Have the interfaces been assessed as to their appropriateness and correctness?

20. Are all administrator and user guidance documents complete and useable?

21. What are the start-up, shutdown, recovery, and rollback policies? Which roles are responsible for these tasks?

22. How is the system delivered, installed, and generated? Which roles do this?
23. What tests are performed in order to ensure correctness? When are these tests done? Who is responsible for conducting these tests?
24. How is the system validated for acceptance and compliance? Who does this?
25. What are the facility requirements, including physical protection of the system? What roles have been assigned responsibility for facility aspects?

5.5 Common Criteria Constraint Conflicts

Returning now to the Shamos constraints, one realizes that even if a system is constructed to be compliant with EAL5 (or any EAL for that matter) there is no implicit assurance regarding any of the fundamental voting system requirements. Further questions must be asked as follows:

I. Thou shalt keep each voter's choices an inviolable secret.

1. What means are used to separate voter identity from voted ballot?
2. How is the balloting process secured such that voter submissions can not be observed, or recorded in any way that is traceable to the individual voter?

II. Thou shalt allow each eligible voter to vote only once, and only for those offices for which the voter is authorized to cast a vote.

3. How are voters authenticated and authorized to cast ballots?

4. What access contro!s are in place to ensure single ballot per voter per election?

5. If multiple systems are deployed, how are voters tracked so the same person does not vote in different formats?

6. What controls are used to ensure that the correct ballot is provided to the voter?

7. What controls are provided to ensure that each ballot item is voted properly?

III. Thou shalt not permit tampering with thy voting system, nor the exchange of gold for votes.

8. How are all forms of tampering detected and prevented?

9. How is vote confirmation provided without ballot-face receipt?

10. How is the voter prevented from retaining a copy of the cast ballot?

IV. Thou shalt report all votes accurately.

11. How does the system assure that each ballot has been correctly recorded?

12. How does the voter know that a cast ballot has been accepted?

13. How is vote tabulation correctness assured?

V. Thy voting system shall remain operable throughout each election.

14. What features are employed to ensure operability throughout the election?

15. How are downtimes handled in the event that they do occur?

16. What alternative balloting system is available for voters when the system is down?

17. How do the poll workers and system administrators know that the system is operating correctly?

VI. Thou shalt keep an audit trail to detect sins against Commandments II - IV, but thy audit trail shall not violate Commandment I.

18. What actions are audited?

19. How is the audit trail accessed and used?

20. What facilities are provided for recount purposes?

21. How is the auditing process precluded from associating voters with cast ballots?

The answers to each of the questions in Sections 5.4 and 5.5 provide assurances about the voting system under scrutiny. Risks will always be present, but a comprehensive security assessment will provide insight to the system's vulnerabilities and threats against its assets.

One issue that deserves particular attention involves the separation of ballots from voter authentication data. As noted earlier (Section 5.2 K.), the CC describes four aspects of confidentiality pertinent to voting systems: anonymity, pseudonymity, unlinkability, and unobservability. The CC refrains, though, from addressing the complexity involved with implementing any or all of these aspects, alone or in conjunction with other secure system entities. Dependencies are indicated between CC components, families, and classes -- saying, in essence, if you implement X then you have to implement Y (and perhaps also Z, etc.). When it comes to counterindications, though, the CC does not provide any similar mapping -- an additional table is needed, showing that if you implement J then you can not implement K (and perhaps also not L, etc.). The fact that conflicting requirements are unaddressed is a major omission of the CC methodology, one that poses a serious problem in voting system implementation.

For voting, there are many such counterindications, but the ones raised by the Shamos constraints begin with auditing (VI) and confidentiality (I). The voter's ballot must be cast anonymously, unlinkably, and unobservably, yet if the system also includes authentication and authorization, the voter identity must not be able to be associated with the cast ballot. Access to the ballot casting modules, though, requires prior authentication and authorization, which could be gained through pseudonymity, although any provided alias must necessarily be traceable back to the voter in order to accommodate the situation where balloting is not able to be completed (either due to system failure or because the voter did not finish the casting process). Pure anonymity and unlinkability are only possible if authentication and authorization transactions occur separately from balloting. Unobservability is compromised since balloting is not transient and must be recorded for recount purposes. The utility of audit trails for detecting and preventing tampering (III) and multiple balloting (II) is reduced by removing an entire class of system users, the voters, from the recorded data. (This is true even if authentication and authorization are not involved, since the balloting sequence must not be retained as this alone could compromise confidentiality.) This impacts the ability to assure operability (V). Since the voter can not confirm that the cast ballot is correctly recorded (due to III) assuring accuracy (IV) becomes a matter of trust in what is an inherently flawed system. This Rube Goldberg chain of calamities would almost be comedic, if it was not also unresolvable.

Chapter 6

I must Create a System, or be enslav'd by another Man's.

-- William Blake

6.0 A Minimal Voting System

Setting aside for a moment all of the issues regarding conflicting constraints, rogue microprocessors, Trojan horse-laden operating systems, corrupted compilers, and spoofed Internet transactions, it would be useful to describe the components of a system that would simply count votes and report the totals, and do no more than just that. The system design could then be examined in terms of aspects of the CC which it would need to satisfy (drawing from the discussion in Section 5.2), and then the Shamos constraints could be applied (as in Section 5.5) to see if any conflicts exist. This would establish a baseline for voting systems, along with an assessment of realizability under a constraint hierarchy, which could be used to appraise other systems as to potential limitations and design tradeoffs. The minimal voting system described here turns out not to be very minimal, since it must include audit trail, tabulation, and display devices that may be difficult to integrate from scratch (without an operating system or device management software). Since all of these devices add layers of complexity and unresolvable auditability, "Ay, there's the rub!" is still embodied in the issue of implementation. At least, though, this description provides a rough-cut way of addressing the salient features of a voting system, those internal balloting items whose security cannot be compromised. The discussion in the next section may be somewhat software-oriented, however, since hardware and software are, in principle, orthogonal, the analogue in a stand-alone hardware-embodied unit dedicated to the voting application could be extrapolated.

A block-diagram of the minimal voting system appears here as Figure 1, and its components are detailed in Section 6.1.

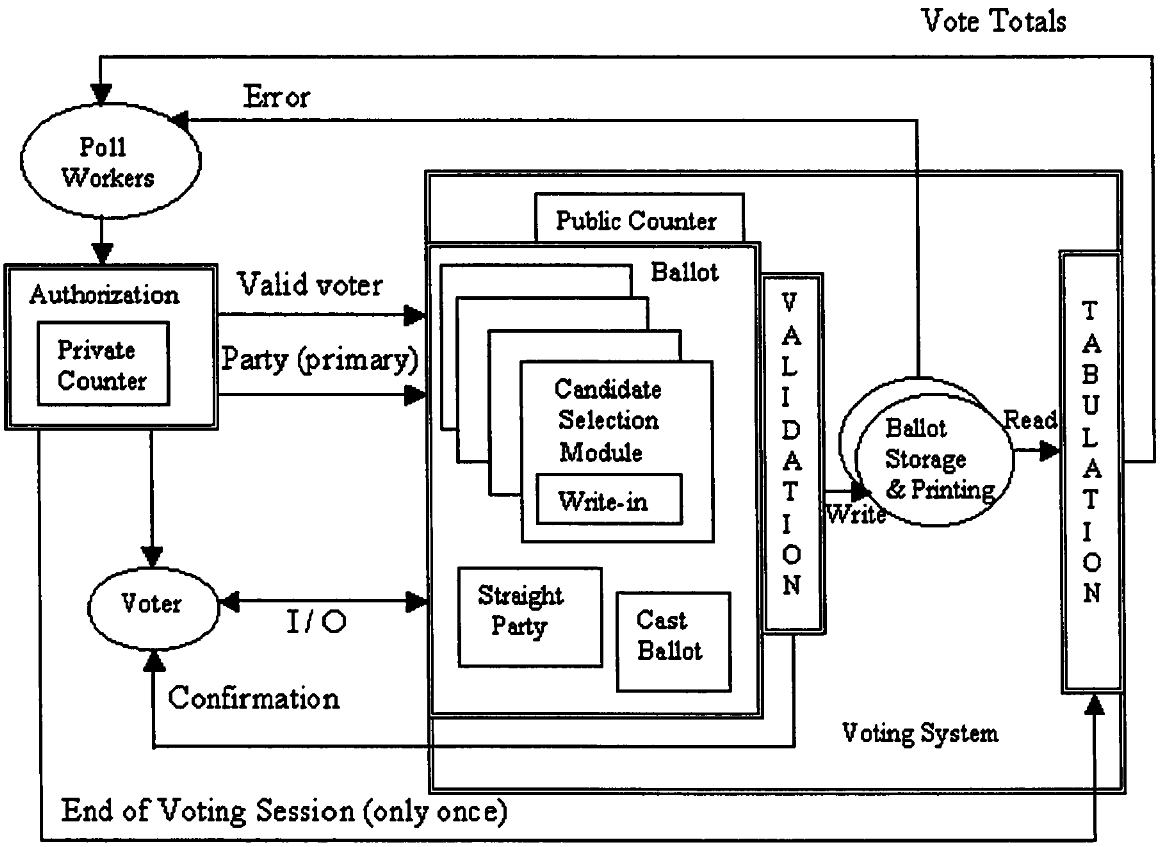

Figure 1 – Block Diagram of Minimal Voting System

6.1 Minimal System Components

The minimal voting system is composed of a balloting section with validation, a write-once ballot storage mechanism, and a tabulation section. There are interfaces for voter and poll worker input and output. The poll workers operate through an authorization mechanism, which is external to the voting system. The system components are described in the subsections below.

6.1.1 Candidate Selection Modules

Interestingly, there are various vote selection types that need to be defined. Each are slightly different in formulation and results. One can think of these as separate modules that can be selected during ballot set-up and preparation. The modules operate independently, although some of their outputs may need to be combined as the inputs to other modules. C++ style pseudocode is used to describe the ballot selection checking methods for ballots that have been cast. Further modules can be defined and added as other candidate selection methods evolve.

A. Two Candidate Selection

A choice is made between candidate X or candidate Y. The voter has the option of selecting either X or Y or neither, but not both. Default values for both X and Y are false.

```
boolean TwoCand(boolean X, boolean Y)
{     return NAND(X, Y);
}
```

B. Multiple Candidate Selection

In an example with multiple candidates, preferential balloting can be used to determine the outcome of the race based on the highest vote recipients. There will be a set of candidates from which a certain number (n) or fewer may be selected. Each candidate can receive no more than one vote. Default values for all candidates are false.

```
boolean MultCand(boolean[ ] X, int n)
{     int total = SUM(boolean[ ] X);
      if (total <= n)
           return true;
      else
           return false;
}
```

C. Weighted Multiple Candidate Selection

A variation of preferential balloting that is becoming increasingly popular in certain types of elections (such as for school boards) permits the voter to distribute a specified amount of votes to one or a group of candidates. Note here that since the selection is not boolean, the integer value allowed to be applied to any particular candidate must not be less than zero nor greater than the total allowed for the group. Default values for all candidates are zero.

```
boolean WeitMultCand(int[ ] X, int n)
{     int total = SUM(int[ ] X);
      if (total <= n)
            return true;
      else
            return false;
}
```

D. Referendum

Referendum issues are formulated as ballot questions to which there is allowed a yes or no reply. This can not be represented as a simple binary selection, since an established default value could be problematic if no choice was made. A ternary value, though, would encompass the no-selection default as well as the yes/no choice. The ternary choice is represented here as an enumerated type.

```
enum ternary {yes, no, nochoice};

boolean Referen(ternary X)
{     return true;     //since enum restricts selections
}
```

E. Write-in

Implementation of write-in is fairly complex. A text area with some reasonable character number limit must be provided for the use of the write-in for all offices except for referendum questions. For multiple candidate selections, there should be as many write-in boxes as there are allowable votes in that office. With a mechanical lever machine, if the voter opens the write-in door next to an office, this precludes selection of a regular ballot candidate for that position, whether or not anything is actually written in the space (of course one could just write a real candidate's name into the write-in spot). If a regular candidate for that office had first been selected, it must be deselected or the write-in door will not open. This same method can be applied to the balloting section of the code, using an elaborate, but systematic, if-else construct. For electronic ballot write-ins, if at least one character is typed in and cast with the ballot, then selection of any other candidate in that office would be voided. It does not matter whether the voter actually writes in a proper name or just gibberish in the text area, if one or more characters appear in the text area, all other selections are precluded. To check the ballot to ensure correctness after casting, the following method would be applied:

```
boolean CheckOffice(string s, office a)
//office is a logical union containing
//  candidate values for that office
//NOCAND is a function which checks that
//  no candidates are selected
{     if (NOCAND(a))     //if no candidates, write-in is allowed
            return true;  //this also accepts no choice at all
      else if (s != NULL)
            return false; //has write-in and candidate selected
}
```

F. Straight Party Vote

Some states allow a straight party vote for a group of candidates via a single location on the ballot. In essence, this performs the office selections for certain endorsed or pre-chosen names. The groups of candidates subject to selection in this manner would be determined and added to the ballot definition, along with an identifying label for each group. Such groups would have to be verified during the set-up process prior to the election, to ensure correctness. Selection of a particular straight party choice by a voter would actually just alter the offices affected by a straight party selection, so the candidates in that group would be selected (and others deselected if necessary). The voter would not be prohibited from using the straight party selection and then going back to change some or all of the candidate selections individually prior to casting the ballot.

G. Vote Changing

Voters should be allowed to change their mind in making selections on the ballot as many times as they like prior to casting their ballot. The ballot is not recorded until cast. (This is typically permitted with mechanical lever machines, but with mark-sense or punched cards involves asking the poll workers to void the incorrect ballot and provide a new blank one.) Changing ballot selections repeatedly or in any sequence should have no effect on the voting system at large, nor should any combination of votes cast have any ability to trigger a system flaw (such as Trojan horse or other bogus code).

H. Primary Election

A primary election requires that the candidates in offices be further separated according to party. Therefore, a particular office would have multiple election definitions, but only one group of candidates for each party. Some municipalities allow candidates for certain offices to cross-file and run in different party primaries, other than the one in which they are registered to vote. The duplication of names within different primaries would still be handled correctly by the triples in the ballot set-up. Primary elections typically require that voters declare themselves to be one (and only one) of the listed parties, prior to (or at the time of) voter authorization. In some states, this declaration establishes the voter's party permanently for all future primaries, and in other states the voter may make their party decision before or at each primary election. Once declared, the voter can not change parties during this balloting session, and is locked-out from voting for any primary office for which their party does not match. An object-oriented method can be used to match the candidate choices to the voter's party in order to confirm ballot correctness. Typically, referendum questions are not asked during primary elections, but if this is required, all voters (even including those who do not declare a party) could access those items (as well as any other non-partisan offices, such as Judgeships) for which the party field was defined as 'none'.

I. Must-Vote Requirement

Some states have a 'must-vote' requirement, in that a blank ballot is considered unacceptable for casting. To further complicate matters, voting just for a referendum question may not satisfy this requirement -- an office candidate must be selected (or write-in performed). Using a mechanical machine, a detected blank ballot will prohibit the voter from using the vote cast lever, and the curtain will remain closed. If the voter still attempts to leave, poll workers would notice the curtains and require that the voter re-enter the booth to complete the process. If the voter refuses to do so, the poll worker would then indicate, in the voting record book, that the voter had not actually voted, and the uncast ballot would be invalidated. With a paper process this checking would not occur -- a blank ballot would have to be accepted since the identity of the voter is lost before the ballot is revealed. With a computer, this check could be performed as a simple sum on all candidate fields cast in valid offices (as previously described for modules A-D, noting that just examining the boolean values does not indicate that a candidate has been selected, since *true* also includes the no-choice selection). As long as the sum is not zero, then a candidate has been selected, and the ballot is deemed proper.

It should be noted here that 'must-vote' is a global (entire ballot) issue, and does not address the matter of undervoting (accidental or intentional skipping of an office or referendum section). This would be picked up at the time of ballot casting.

J. Ballot Casting

When a voter is done making selections, a casting operation would be selected, which should issue a feedback message ("are you ready to cast your ballot?") with confirmation to complete. This would assist in protecting the voter from accidentally submitting a ballot, thus ending their voting session, before they had intended to finish. Incorrect ballots can be prevented from being cast by performing checking prior to submission, again giving the voter a prompt to correct their ballot if it is wrong (overvoted or blank) or incomplete (undervoted). The system could be set up to not allow the casting session to end if a ballot is wrong. Alternatively, the system could permit casting and then the office(s) for which an improper vote is flagged would be voided, just as would occur with a paper balloting system. Undervotes should be permissable, but feedback about skipped choices will alert voters if this happened inadvertently.

An essential part of the ballot casting process involves printing a paper record of the votes cast. Once a voter confirms that all choices have been made, the system produces a written record of the ballot which is then shown under a transparent, photograph-resistant, tamper-proof screen. When the voter indicates (through an additional selection) that this is an accurate display of the choices made, the system drops the printout into a secured ballot box (as described in Section 3.3 of this thesis). A procedure involving poll worker alert should be in place, if a voter finds that the printout does not reflect their choices correctly.

6.1.2 Ballot

The ballot would be described as a class containing:

1. Candidates listed as (name, party, office) triples, where if the office is of type Referendum, the party can be set to 'none' (there the name would be the context of the question). The party field (whose values would be defined in an enumerated type) can also be set to 'none' in elections where no party is declared, or when a candidate is allowed to have no party affiliation (which is different from 'independent' -- an actual party in some states).
2. Elections listed as (office, type) duples, where the election type is from modules A, B, C, D in Section 6.1.1.
3. Checking methods would be assigned according to election type.
4. Each voter's ballot data space would contain a value for each candidate, initially all set to the defaults.

For candidate selection, the boolean result of false indicates that the ballot was improper for that office module, a result of true indicates that it was correct. These results could be used to provide feedback to the voter in order that they may correct their ballot. Note that proper ballot set-up (candidate triple definitions and so on), although an essential part of pre-election data checking, is assumed to be correct for purposes of this vote checking discussion. Data errors here would be visible to the voters, and they could report discrepancies, but this is not a substitute for a strong pre-election validation process.

6.1.3 Authorization

The minimal system is proposed for use at traditional polling places where voter authentication would be performed by poll workers, who would (as is already commonly done) allow only qualified voters to proceed to the balloting area. There, prior to each voter's use, another poll worker would visually check that the machine is cleared and ready for a new ballot. This may include a small procedure that must be performed before the voter is authorized to use the machine. The procedure might involve keypresses on a separate input device that the voter can not access, in much the same way that a special lever is used on the side of a mechanical voting machine, to prevent double-balloting. For primary elections, the poll worker also performs the party selection for each voter. Poll worker selections must also have no way of affecting the ballot data contents or tallies.

6.1.4 Ballot Storage

If strong assurances are provided that voter and poll worker transactions are prohibited from having any adverse effect on the voting system, it should be unnecessary to record each individual keypress during the voting process, as would be done in order to keep a full audit trail of the election. A full, consecutive, audit trail has the potential of violating ballot privacy, and a randomized audit trail is not an audit trail, so other methods of auditing should be used. One possibility is to retain each of the ballot images as cast, but to place them in randomized locations on a write-once device. This would necessitate some confirmation from the voter that the ballot cast and then permanently recorded is what they submitted, a step that could cause delays and confusion. The method suggested here is to provide a set of full ballot record locations that have all been pre-initialized to the default votes (which should be no-votes in all offices), enough to hold the maximum number of ballots that could be submitted during the voting session. As a voter begins their session, they are randomly assigned to one of the pre-initialized locations, which allocates their ballot for use. The voter uses this memory area to make their choices directly into the memory locations that will be used for the audit trail. Casting the ballot locks this particular ballot record area and protects it from further use, deletion or modification. This could be done using a read-write CD-ROM or EEPROM with some additional hardware to perform the locking function. A check-sum may also be provided along with the cast ballot, to additionally assist in flagging any accuracy or integrity problems. Note that this discussion of audit trail addresses only the issue of ballot auditing. For all other operations (set-up, shut-down, etc.), a full audit trail should be kept in write-once memory, since privacy is not a concern with these transactions. The paper ballot records must also be secured, along with the voting system, for tabulation purposes.

6.1.5 Tabulation

Various types of totals need to be generated by the voting system. For example, many municipalities maintain a public counter and a private counter on each voting machine. The public counter is a running total of the number of ballots (not individual votes) cast on the machine during that particular session. The voter should be able to see this number during the voting session. In no way should the public counter numbers be able to be traced, or have any relationship to the memory locations where particular ballots are recorded. The private counter is a tally of the entire number of ballots cast on that machine during its lifetime. This is traditionally used, in conjunction with the separately maintained public counter, to double-check the number of votes cast during the election session. The private counter may be unnecessary in a software-based system, but it could have utility in a hardware-based machine format.

The individual votes cast for each candidate or referendum choice can be kept as a running tally as each ballot is cast during the course of the election. These totals, kept in non-volatile (read-write CD-ROM or EEPROM) memory can be used as a double-check against another set of totals that can be generated (off-line) after the election by re-tabulating the ballot images. The area used for the running totals must be properly initialized during set-up, and there should be no way to decrement any of the counters or modify them through any process other than casting a ballot during the course of the election. The running totals should not be readable by voters, poll workers, or system administrators during the interval when the machine is available for balloting. Following the election, these counters would be frozen and revealed (through print-out or display) to provide the returns. It is important that accuracy and integrity be maintained for the tallying system, and that the totals conveyed properly reflect the sums generated by ballots cast.

The electronic ballot totals should be used only for preliminary result reporting. Although candidates may wish to concede the election on the basis of these totals, there should be a second set of results produced directly from the paper ballots. If there is any discrepancy between the electronic and paper ballots, the paper results should be used, since this is actually what the voters confirmed as being representative of their final choices. The paper can be read via optical scanners (using multiple systems for checks and balances, as discussed in Section 5.2.L on recounts) or even by hand.

6.1.6 Administrative Tasks

There are certain administrative tasks which must be performed by the minimal system as follows:

1. Start-up at the beginning of the voting day should include a checking procedure that verifies the system's correct operation. This can be automated but may also include some manual processes.
2. The system should have a way of sending an alert if a malfunction occurs. It is especially important that the inability to write ballot data out and secure the voted record be detected and reported.
3. The shutdown procedure, which includes reporting the vote totals, should also include re-verification of the system using automated and possibly also some manual processes.

All set-up, shut-down, and alert processes should clearly indicate whether or not the system is able to be used for voting. If malfunction occurs, it should not be possible to continue to accept ballots on the system, until it has been serviced and returned to proper functional operation. These tasks are in addition to any others which would be performed prior to the election, such as installing the ballot information and other preparation activities.

6.2 Minimal System Realizabililty Under Constraints

Let us now examine the proposed minimal system, under the Shamos constraints (keeping in mind the questions asked in Section 5.5), while considering at least some of the implementation issues.

I. Thou shalt keep each voter's choices an inviolable secret.

If the units are stand-alone and electromagnetic emissions are shielded, external monitoring should not pose a problem. Networking, of course, would offer data channels that are less controllable; hence, it is not recommended. The Internet is even worse than a closed network, as previously discussed, so it should not be employed for voting at all.

As far as internal secrecy is concerned, the use of a randomly assigned memory area for each ballot would constrain each ballot's data to that location. Scratch memory used for ballot checking could also be constrained to the voter's data segment. The screen memory may be involved for displaying the ballot face, but this would need to be cleared, along with any other related memory which pertains to the ballot, immediately following vote casting. The integrity of these clearing operations must be ensured.

As well, there should be no areas of memory (volatile or non-volatile) containing ballot data that could be traced back to the sequence of ballots cast (viewed externally by listing the order in which voters used the machine). These clearing functions (as well as clean installation of the voting product) may be difficult to assure, hence the computers used for voting should not be accessible for general (non-election) use.

II. Thou shalt allow each eligible voter to vote only once, and only for those offices for which the voter is authorized to cast a vote.

Note that the minimal voting system does not include any automated voter authorization process. This must be done manually, because there is no way to ensure that authorization data remain separate from ballot data, due to the necessity of maintaining audit trails for both sets of processes. As discussed earlier, any system that requires both authorization and anonymity puts serious demands on the controls separating these processes.

Access to the voting system would be monitored by the poll watchers, as described earlier. Casting of additional votes by the poll watchers would be detected, because the public and private counters would not concur with the number of voters that were authorized. The poll workers should be representatives from different political parties so collusion is better controlled. The voter signature requirement provides additional checking against fraud.

To ensure proper voting, the remaining ballot checking would be performed by the system using the methods detailed.

III. Thou shalt not permit tampering with thy voting system, nor the exchange of gold for votes.

Since the voters do not retain a receipt that shows how they voted, and only the fact that they did vote is noted by the authorization process, the matter of exchange of gold for votes would be no different from other voting systems. Certainly a voter could take a photo of the electronic ballot face (not the printed ballot, since it is displayed under a photo-resistant screen), but there would be nothing to stop the voter from changing the settings after taking the photo (and then casting the revised ballot), which is the same as with current methods.

Tampering with the system would be avoided by not allowing the voter to have direct access to the ballot readying components, and by segmenting all voting operations and data such that there is no possible access provided to the voter to system internals. This would need to be clearly demonstrated in the system design and implementation.

IV. Thou shalt report all votes accurately.

Hardware errors can and do occur, but these would be minimized through the use of lockable non-volatile memory systems to provide a redundant vote count, along with checksums to detect inaccuracies or anomalies. The paper ballots provide the ultimate arbiter for the votes cast, and these can be re-read if election results are contested.

The minimal system simplifies the ballot verification process so that it can be contained in a small amount of code or hardware that would be subjected to scrutiny and verification during system acceptance testing. These critical areas should even be required to be open systems components, to enhance credibility of function.

V. Thy voting system shall remain operable throughout each election.

The non-networked system would be less susceptible to denials of service attacks or inaccessibility due to communications breakdowns. Hardened casings and other physical barriers from abuse could be employed in the construction of the units. Alternate power backup should be available in case of outages or malicious unplugging.

The fact that the system uses non-volatile memory for its ballot records, audit trail, and tallying, would mean that even if the system goes down in the middle of an election, it could be replaced with another functional unit and then the information contained in the malfunctioning unit could be extracted and added manually to the totals from the replacement machine. This is similar to existing lever machine operations, which also occasionally fail, but retain their vote totals regardless. The memory systems must be protected from power surges, magnetic interference, and other destructive influences.

The poll watchers would continue to provide the service of protecting the machines from overt attacks. The segmentation of access to the system away from the voter also protects operability.

It is important that there be no reliance on real-time clock settings for performing any voting related function, as this provides an area of high vulnerability.

VI. Thou shalt keep an audit trail to detect sins against Commandments II-IV, but thy audit trail shall not violate Commandment I.

The audit trail and ballot retention system described should provide such protection.

An implementation of the minimal system would also require an assessment of security using the questions outlined in Section 5.4.

6.3 California Internet Voting Task Force

It is illustrative here to discuss the report issued early in 2000 by the California Internet Voting Task Force.[CAI00] This report deserves consideration as the most recent extensive examination of the electronic vote tabulation subject from a government agency, and also forms a point of comparison for many of the issues raised elsewhere in this thesis. (Quotations in the body of this subsection are taken directly from the Task Force report.)

The State of California takes pride in its high-tech nature, claiming to be the birthplace of the Internet, hence it is not surprising that they should be considering a move from their paper-based, electronic tabulated voting systems to an on-line format. An Internet Voting Task Force was formed by California Secretary of State Bill Jones, in order to study the feasibility of implementing Internet voting (or I-voting). Members included representatives from the Secretary of State's Elections Division office, elections-related officials (from State Senate and Assembly Elections Committees and various County elections offices), public interest groups (such as the League of Women Voters and the California Voter Foundation), educators (from Cal Tech, San Jose State, and UC Riverside), and numerous industry representatives (including Cisco, Oracle, Global Elections Systems, VoteHere.net, FAQvoter.com, and others). The Task Force produced a Technical Committee Recommendation Report which was issued along with their

Feasibility Study Report in January 2000. Additionally, a draft standard for certification of certain Internet voting systems was issued on February 14, 2000.

The resulting recommendation of the Task Force was that Internet voting should be provided as an additional balloting method in California, and that it should be deployed through a four-stage evolutionary process, in order to give time for voters and officials to get used to the system, and for developers to address technological issues. Systems would remain based within the individual California counties, for administrative purposes, and the approach used would be similar to the current paper absentee ballot system, in terms of authentication of the voters and their voted ballots. The initial two stages of deployment would involve the use of county controlled Internet voting machines, first only at the regular polling sites during election day, and in the second stage at alternative voting sites staffed by election officials for days or weeks in advance of the election. The second stage would permit voters to access their proper local ballot from any Internet voting machine within their home county. The third and fourth stages of deployment would require the implementation of an authorization process enabling a voter (who has previously requested to vote via Internet) to cast a ballot at an unattended voting kiosk (situated at public locations, such as libraries), or in the final stage, from almost any Internet accessible computer.

The report examines many of the known technical, sociotechnical, and sociological issues related to computer-based election systems, and makes a good case for formulating their proposed remote I-voting scenario on the existing absentee ballot system. It describes well the transactions that would be required in order to authenticate a remote voter and his or her voted ballot, and also details many major security vulnerabilities that must be addressed prior to any system certification. The conclusion appears to be that systems which are under the control of the county could be enabled for I-voting as described in stages one, two, and three, but that the use of privately owned personal or office systems for remote voting in stage four poses serious technical problems and is not recommended for consideration at this time. The fact that despite this recommendation, the stage four description persisted as a plausible eventuality in the feasibility report should be of some concern, since it leaves the impression that a solution to these problems is possible and forthcoming, which is not necessarily true.

Since California does have a long history with computer-based vote tabulation (primarily via punch-card and mark-sense systems -- which are manually auditable, as has been occasionally performed in close races there), the overriding issue of whether or not computers should be relied upon at all to collect and count votes seems to have been considered irrelevant (for their purposes, the open questions regarding this matter not withstanding). As for Internet voting, though, the Task Force's report might be used as a good vehicle for furthering open discussion, but as a feasibility document, it is seriously flawed. Four areas are particularly salient, as follows.

I. *The Internet does not presently provide a secure conduit for any aspect of the election process.*

Given that any Internet-based system carries with it the panoply of problems (cited in Section 2.5 of this thesis) that could have serious impact on an election process, especially those which the computer industry has not yet been able to fully resolve (such as denial of service attacks and firewall breaches), it was therefore inappropriate for the Task Force to assert that any of the proposed I-voting stages could "provide at least the same level of security as the existing voting process." If the intention of stages one and two was to provide direct access to the county voting system from terminals at polling places or alternative sites, one must question why the Internet was recommended for use, rather than a direct-dial wide-area network? The only plausible reason for Internet use in stages one and two (other than the fact that software and systems are readily available, although extensive modification to add security features is likely necessary) is as a preparation for stages three and especially four, where the Internet provides more convenient access. On the other hand, even stage three could be implemented using direct-dial since the county would know in advance how many simultaneous connections to anticipate, and the server could be configured appropriately. Stage four would need to accommodate voters using Ethernet and other non-phone systems, but the Task Force deemed that such networks would be problematic, for other reasons (like firewalls and monitoring in office environments) anyway. Direct-dial for stage four would allow out-of-state absentees to cast I-votes, although long-distance charges would be incurred unless a toll-free number was provided.

The eventual trend is probably the elimination of dial-up modems as cable and other technologies offer higher-speed alternatives, so perhaps one could understand why the Task Force thought the Internet might be more appropriate than direct modem access, since stage four implementation was anticipated further into the future. But as illustrated by New York's procurement process, new voting systems tend to be costly and hence are intended for long-term use (certainly at least a couple of decades). Since the Internet is market-driven and essentially unregulated, its future composition is difficult to predict -- any Internet-based solution devised today also brings with it the possibility of rapid obsolescence, a serious problem that could inadvertently result in squandering the allocated public funds for all or part of the I-voting project.

II. ***Encryption technology and the use of secure channels can not be relied upon to ensure ballot integrity and secrecy.***

Certain aspects of the described I-voting implementation rely on the naive assumption that integrity and secrecy are guaranteed over the Internet since ballots would be encrypted, according to "current professional standards" prior to transmission. This is patently untrue (as discussed in Section 3.5). Without fully securing the vote system end-to-end, not only during distribution and use but also by monitoring the actual development of the balloting software (an issue which is not adequately addressed in the Task Force report) to ensure that trap-doors or other bogus code is not inserted into encryption or other processes, ballot integrity and secrecy should not be assumed. Neither does the use of secure channels eliminate denial of service, re-routing and spoofing issues (see Section 3.7).

The proposal describes a feedback mechanism that provides notice that a voter's ballot has been properly received. This could certainly be fraudulated, most easily at the unattended kiosks where look-alike software could be installed to provide a misleading reply. It is not clear how to check that all kiosks continue to be operational throughout the election -- non-receipt of ballots might indicate a malfunctioning terminal, or just a lack of voter population in that particular polling location. Each individual voter could (following the election) check to see that their ballot was actually registered (how many people actually would?) -- but since (for matters related to vote-selling and privacy) the contents of the voter's ballot can not be re-accessed, it is impossible to verify its correctness. Furthermore, there is no correction process for a voter who presents notice of receipt and then discovers that the system did not record that their ballot was actually cast. In this case, the only recourse would be to contest the entire election, since allowing re-voting is obviously precluded.

III. *Internet voting is not equivalent to or an improvement over paper absentee balloting.*

Stages three and four address the matter of voters who would be disenfranchised due to an inability to be at their local polling place on election day. As with paper absentees, if it is known in advance that this will be a problem, an Internet ballot authorization could be requested. The described authorization process involves submission of a signed I-voting request by mail (since digital signatures or biometric identification methods are not yet universal, nor are they necessarily secure), with the voter receiving back by postal mail an authorization password, instructions, and possibly also access software on CD-ROM.

Clearly, it is far easier to send the voter a paper ballot that can be marked and returned in a matter of minutes, than it would be to require the voter to re-boot a home or office computer in order to install "a clean, uncorrupted operating system and/or a clean Internet browser" to secure an Internet voting machine prior to use. The report describes I-voting as helpful to "the occasional voter who neglects to participate due to a busy schedule and tight time constraints," but until the process is reduced to a few browser clicks, one should instead assume that these people will likely continue to remain uninvolved. Indeed, the report even states that "additional complexity is the inevitable price of security and convenience," yet in another section it is noted that "Internet voting systems shall be user friendly and offer voters a simple, convenient and uncomplicated opportunity to vote." This conflict is unresolved.

In actual fact, though, when the Alaskan Republican party provided (for their January 2000 presidential straw poll), an Internet voting system by VoteHere.net that required software installation, only 1% of the people eligible to vote in that fashion cast ballots, despite the fact that this is a state where traveling to the polling place is problematic.[LED00] Somehow this information did not manage to make its way into the report, even though VoteHere.net was one of the Task Force members.

Where I-voting possibly offers any advantages at all over traditional paper absentee balloting is in checking the ballot prior to casting in order to minimize ballot loss due to spoilage (improper voting, say for too many candidates in a category), and in eliminating postal delays. Tabulation is no quicker, since paper ballots can be computer scanned, and even the spoilage and loss problems could be alleviated through better instructions and ballot design, along with provision for additional, secure, early drop-off locations.

IV. *The Task Force report and draft certification standards fall short of a secure system specification.*

Although standards for development and evaluation of secure systems (such as the recently developed ISO Common Criteria, and TCSEC, its predecessor) exist and are required for certain critical government computer systems, there is no suggestion at all in the Task Force report that such criteria be applied to the assessment of all or part of any I-voting systems. Yes, California does have existing voting system standards, but no, these standards do not include the more stringent CC standard for secure systems.

The use of a Technical Review Committee and Independent Testing Authorities for certification (as recommended by the FEC guidelines) is ad-hoc at best, without the establishment of minimum security benchmarks against which the review and testing must take place. The ISO standard indicates, for example, the complex nature of stripping authentication information from a piece of data that is to remain anonymous, and it requires in-depth system review in order to ensure that it is impossible to re-associate the two groups of data -- this disassociation process is what the California proposal says the ballots will undergo, but there is no indication that the Task Force comprehends the difficulty in providing assurance that this actually happens inside of the vote tabulation system.

The absence of any mention of well-accepted security standards (such as the CC or GASSP) is a serious omission from the Task Force report, as is the additional omission of the impact in terms of cost and time for a thorough certification process. The report focuses mainly on the user interface portion of I-voting, the voter verification process and how voters will access the balloting application, but deals hardly at all with the far more critical functioning of the system that actually will collect, record, and tabulate the votes. The central system needs to be well-specified and subjected to rigorous acceptance testing and examination, yet this non-trivial matter is not examined by the Task Force.

In short, we are left with various questions regarding the California Task Force report. Why is the Internet being used at all when a closed network could be more safely deployed for stages one, two, and three? How will ballot integrity and secrecy be ensured since encryption alone is inadequate? Why is a cumbersome and possibly expensive process being added to the state's approved set of voting systems? What standards will be applied in order to review and certify the new systems? In light of these open matters, one would tend to agree with the Task Force's statement that "if Internet voting is viewed skeptically...then the fundamental trust in the democratic process may be compromised." It is therefore incumbent upon the Secretary of State to continue the discussion regarding Internet voting in order to address these critical issues prior to any consideration of entering the stage of accepting I-voting system proposals for review.

Chapter 7

Securus iudicat orbis terrarum. (The verdict of the world is conclusive.)
-- *St. Augustine*

7.0 Conclusions and Recommendations

This thesis has stated that electronic vote tabulation systems can be created, but not without a host of insurmountable problems. Electronic vote tabulation systems necessitate stringent and elaborate development, assessment, validation, and maintenance procedures, most of which require a high level of technical expertise, typically well beyond that which is currently used by overseeing election administrators. Voting system security must be all-encompassing, and will only be as strong as its weakest links.

Computer-based voting offers the promise of easy access and speedy tabulation in exchange for a variety of risks that were either not present or are far worse than ones found in manual balloting systems. Some problems, such as those involving large-scale fraud, denials of service, and the incompatibility of anonymous balloting with audit trails, are inherently unresolvable. Vendor monopolies should be prevented in order to reduce globally propagated errors or product-specific attacks. Certain technologies, such as Internet voting and remote voter authorization, are particularly vulnerable to these risks, as well as other sociological problems (like vote selling and coersion), and should not be used at all. Wherever possible, voter authentication, ballot casting, and vote tabulation should be provided in separate systems. DRE-style devices may offer the best potential compromise for vote casting, since they can be outfitted to provide human-verifiable printed ballots that can be used for recounts, and can be otherwise self-contained and security hardened (although considerable work remains before any current models would satisfy the criteria discussed in this thesis).

Electronic vote tabulation can never be viewed in isolation as just a computer application problem, since the inherent risks of automation will always be further compounded by the adversarial nature of elections. Purely technological solutions fail to adequately address the sociological issues that are part and parcel of the democratic election process. Voting systems therefore must require the inclusion of human checks and balances as a necessary implementation component.

The future will likely hold continuing improvement in overall security techniques, including evolving standards for system development, assessment, practices, attack and misuse detection, authentication, and other aspects relevant to vote tabulation. Research efforts will lead to more secure compilers, automated analyzers, operating systems and computing environments. At the same time, nefarious mechanisms to thwart these enhanced systems will probably also continue to advance. The analogy that the atomic bomb did not (as some had hoped) eliminate all wars, holds true here as well, since undesirable human behavior continues to persist (and eliminating that carries with it other unfortunate Orwellian consequences). Still, there are some efforts that are worth pursuing in the electronic vote tabulation context.

The Federal Election Commission (and its analogue in other countries that deploy electronic voting systems) must be charged with the creation of minimum standards to be applied to any hardware or software used in U.S. national elections. These standards should be concrete, tracking existing security documents, so that they can be readily applied. These minimum standards should not impinge upon states' rights, as each municipality should still be able to impose or create additional criteria and requirements. The argument that mandating stringent security would make the resulting systems too expensive for municipalities to afford would be offset by providing a uniform set of criteria, along with recommendations as to how such could be implemented.

In much the same way that NIST's engineers are used to certify the concrete used to build our nation's infrastructure, so too should the computer security experts at NIST be employed to secure the structural integrity of our democratic elections. Using the Common Criteria, NIST could formulate testing standards for voting systems, thus allowing vendors to develop their unique systems from an established security base. Where the Common Criteria is inadequate for voting system assurance, additional procedures should be applied. NIST should also be used to design an evaluation mechanism that could be applied by purchasers in order to adequately assess the evaluation reports provided by Independent Test Examiners during the procurement process. By shifting some of the funding and establishment of requirements and assessments onto the appropriate Federal agencies, and by providing a template for voting system development, the overall costs should decrease, while improving the security assurance in the resulting products.

Federal, state, and municipal legislative bodies must mandate the development and use of these standards, and also take a strong stand against the use of highly flawed technologies in voting. Other laws, such as those pertaining to malicious attacks and vote-selling should be tightened, and prosecution of those who violate these laws must be vigorously pursued.

It is also essential that voting system vendors be held accountable for their claims, so that the public knows what to expect (and what not to expect) from their products. It is not acceptable to simply state that a system is secure because it uses encryption or provides some sort of audit trail. The vendors should be held to task to rigorously explain the features of their voting systems, in lay-person terms. Developers of voting systems should not be permitted to hide behind trade secrets, and must fully disclose their systems to appropriate agencies to allow comprehensive testing and evaluation. The voters deserve to know when certain voting criteria can not be satisfied by the systems they are using, and they should be allowed to "opt out" and vote with an alternative non-electronic method, if their right to an anonymous, secure election is compromised by their district's election equipment.

The system guidelines provided in this thesis could be used as a framework for the creation of an actual voting system that would be subjected to extensive testing. Non-partisan agencies concerned with the preservation of the democratic process should provide grants to fund multiple projects of this nature, so that this technology can be advanced through independent research.

In conclusion, it is probably unfortunate that the groundswell of enthusiasm for electronic vote tabulation can not be abated until such time that appropriate assurances of accuracy, integrity, and anonymity in elections can be provided along with significant computer-related risk minimization. Yet, one can still hope for fairly administered democratic elections as long as citizens demand and confirm, with the best of their abilities, that appropriate checks and balances continue to always be applied.

Chapter 8

The conduct of a losing party never appears right: at least it never can possess the only infallible criterion of wisdom to vulgar judgments -- success.

-- *Edmund Burke, 1791*

8.0 Postscript

Eleven days after this dissertation was defended, the United States held a Presidential election. In the days, weeks, and months that followed, many of the experts cited here (including myself) were called upon, by the legal teams (who were defending or protesting the Florida recount) and the news media, to provide expert testimony and commentary on the vote tabulation process. Hearings have continued to unfold, as legislators and municipalities examine their current voting practices and rush to replace equipment that they fear could cause "another Florida" in their districts. Although much of what was revealed in the court cases and news articles was "new" to the general public, it was not at all a surprise to those who have been on the front line of election investigations over the last decade or more. Election Data Services' president Kimball Brace's words in the Los Angeles Times July 1989 series of articles (cited in Sections 2.3 and 3.2) seem almost eerily prophetic now:

> "The election community is in a state of potential crisis." "We're waiting for a volcano to erupt, in the form of a major election scandal . . . We know it's going to happen, but we don't know when or how we're going to handle it."[TRO89c]

While punch-cards and chad took center stage in the courtrooms, voters learned that, in fact, every vote does not count. Shamos' assertion regarding accurate vote reporting being low on the list of election commandments was certainly true. The discarding of nearly a quarter-million undervoted ballots in one state alone now seems shocking, but it is not news to most election officials (there or elsewhere). Back in 1988, some 210,000 votes were not recorded for the U.S. Senate race in Florida's Dade, Broward, Palm Beach and Hillsborough counties (sound familiar?). These happened to be the four heavily Democratic regions which were then said to favor candidate Buddy MacKay, who lost the election to Republican Connie Mack by some 35,000 votes statewide. Some statistics from that election bear examination:

> "In a comparable presidential-year U.S. Senate election in these four counties in Florida in 1980, for example, out of every 100 Floridians who voted on the presidency three did not vote for a Senate candidate. But in 1988 in the same four counties *fourteen* out of every 100 citizens who voted for President were not recorded as voting for either MacKay or Mack. Comparing 1980 to 1988, in Broward County the Senate-race dropoff increased from 3.6% to 6.8%; in Dade County, from 4.2% to 13.1%; in Palm Beach County, from 1.1% to 16.5%; and across the panhandle in Hillsborough, from a flat 1% in 1980 to a whopping 24.5%

> in 1988. Thus in Democratic Hillsborough County one in four voters for President disappeared from the voting totals for Senator."[DUG94]

As in 2000, ballot layout was suggested as a possible cause for the excessive dropoff. Dorothy Joyce, then Florida's elections chief, was quoted as saying: "We thought all along it was purely the ballot layout that was the problem. I still believe that's what it was. . . . It was just very obvious." But another explanation was proffered by MacKay:

> "My understanding is that it's possible to program a computer for it to count wrong for a while and then straighten itself out. Now that I realize the level of sophistication, I realize how easy it would be for somebody to in effect rewire that program for a brief period of time and then have it straighten itself out. I don't know how anybody could prove that." Rising as the interview ended, Lt. Gov. Buddy MacKay said "I just hope it doesn't happen to somebody else."[DUG94]

The transcript of testimony given by Jerry Williams, programmer for CES, the corporate predecessor of the vote tallying system supplier Cronus/BRC (at a deposition on July 14, 1988 on this matter) indicated that a COBOL subroutine "identified by the letters 'CRT-RTN' was embedded in the 'EL-80' votecounting source code which was in use in Hillsborough and Pinellas counties, Florida." "Hillsborough County election officials had used it, he said, for a subroutine which enabled the display of cumulative results on remote video terminals, but he conceded that 'CRT-RTN' could become any kind of computer code that a programmer wanted to turn it into. All a programmer had to do, Williams said, was key his chosen subroutine into the source code at the line for 'CRT-

RTN' and blank out an asterisk which stood at the leftmost edge of the line, and at that point in the program, it will be executed." What is highly disconcerting is that the 2000 Presidential dispute focused primarily on recounts of ballots that were identified as undervoted (due to chad problems) and the underlying source code issues (including those programs used to separate the ballots into tallying piles) were largely ignored.

Nor is disenfranchisement of voters by virtue of defective equipment a new issue. Federal Judge William L. Huntgate ruled in 1987 in a election dispute in St. Louis "that the computerized punch-card voting system as it has been used in that city denies blacks an equal opportunity with whites to participate in the political process." In the contested election, "voting positions on ballots in the black wards were more than three times as likely not to be counted as those in white wards."[DUG88] Some believe that proper ballot preparation may be tantamount to a modern-day literacy test. Instructions for use of the punch-card devices were printed on the Florida machines, thusly:

> "AFTER VOTING, CHECK YOUR BALLOT CARD TO BE SURE YOUR VOTING SELECTIONS ARE CLEARLY AND CLEANLY PUNCHED AND THERE ARE NO CHIPS LEFT HANGING ON THE BACK OF THE CARD."[US00]

But since the cards were not printed with the names of the candidates, it is highly unlikely that a voter would be able to determine, through a visual perusal of their punched ballot, whether or not they had prepared their card correctly (see paragraph 11 in the Appendix of this thesis). Other known suggested fixes that could have eliminated many of the card problems, such as spring-loaded punching styli and card readers at the polls, have not been implemented.

Ultimately the U.S. Supreme Court deemed the 2000 Presidential election a states' and municipal rights issue (as discussed in Section 3.1 of this thesis). A community has the right to cast ballots and count votes in the manners in which it sees fit. What it does not have the right to do, in the opinion of the Court, is to change the rules of the game in mid-stream, no matter how poorly these rules have been written and applied in the first place. The per curiam decision read (in section IIB):

> "The question before the Court is not whether local entities, in the exercise of their expertise, may develop different systems for implementing elections. Instead, we are presented with a situation where a state court with the power to assure uniformity has ordered a statewide recount with minimal procedural safeguards. When a court orders a statewide recount remedy, there must be at least some assurance that the rudimentary requirement of equal treatment and fundamental fairness are satisfied."[US00]

Yet the same equal treatment and fairness apparently does not apply to the procurement of voting systems, as the Caltech/MIT Voting Project revealed. Their study compared the undervote (which they refer to as residual vote) rate by machine type in U.S. counties in the 1988-2000 Presidential elections and concluded:

> "Paper ballots, lever machines, and optically scanned ballots produce lower residual vote rates on the order of one to two percent of all ballots cast over punch card and electronic methods over the last four presidential

> elections." "The incidence of such residual votes with punch card methods and electronic devices is forty to seventy percent higher than the incidence of residual votes with the other technologies."[CAL01]

Evidence against the reliability of DRE machines continues to mount. A Republican poll worker in South Brunswick, NJ noticed, during their November 2000 election, that a brand-new kiosk-style electronic voting machine failed to register votes for two major party (one Democrat and one Republican) candidates in a county freeholder race. When the manufacturer was asked why their supposedly fail-safe system failed, their only explanation was that "votes were never cast."[GEL01] The distinction between the intention of a voter to cast a ballot, and the requirement that a voting machine properly record the choices, will continue to remain an issue as long as system vendors are not held accountable for the accuracy and reliability of their products.

Voting should not be akin to a survey (poll), where margins of error are expected. Every intended vote should be countable and counted, or if not, then election race tallies falling within the 'gray area' must be determined a 'dead heat' (as in any other scientific use of survey material) and re-elections must be scheduled and conducted until statistically significant results are obtained.

The U.S. Supreme Court wrote: "After the current counting, it is likely legislative bodies nationwide will examine ways to improve the mechanisms and machinery for voting."[US00] The numbers of new bills introduced in Congress and in the state legislatures at the end of the 2000 and the beginning of the 2001 sessions indicate that there is now considerable interest in voting system reform. But many of these pieces of proposed legislation show a serious lack of understanding of the numerous and often conflicting sociological, technological, and sociotechnical matters described in this thesis. The prevailing mode appears to be to "buy new equipment now, so we look like we're doing something about the problem, and we'll worry about standards later." The FEC, rather than being funded and charged with establishing new voting system criteria, is being considered for abandonment or absorption into other election policy bodies. The existing NIST security assurance program continues to be ignored by vendors and procurement agencies alike. A host of new voting devices ranging from palm pilots to punch-card retrofits are being rushed to market, and uncertified products have appeared in 'dog-and-pony-show' technology demonstrations in Washington D.C. and state capitols around the country. All of this is appalling.

The nation seems ready to jump out of the voting system frying pan into the fire. It is my sincere hope that sanity and science will prevail, and that this document as well as the handful of other new serious writings on the subject will be read by the decision-makers who are purchasing and designing the election equipment of the future. A government that is "by the machines, of the machines, and for the machines" can scarcely be called a democracy. Constant vigilance must be used to ensure that this does not happen to our election process.

Appendix

The following document is the sworn affidavit I was requested, by the Democratic Recount Committee, to provide regarding the necessity of a hand recount in the disputed Florida precincts for the 2000 Presidential election.[MER00] The testimony was presented as part of the defense brief in the 11th Circuit Court of Appeals, Atlanta, November 17, 2000. Reference to this affidavit was made in the "Brief in opposition of respondents Al Gore, Jr. and Florida Democratic Party in Nos. 00-836 and 00-837" presented by Laurence H. Tribe to the United States Supreme Court on November 23, 2000.

UNITED STATES DISTRICT COURT
FOR THE SOUTHERN DISTRICT OF FLORIDA

CASE NO. 00-9009 CIV-MIDDLEBROOKS/BANSTRA

DOUGLAS, GONZALO DORTA, CARETTA
KING BUTLER, DALTON BRAY, JAMES S.
HIGGINS, and ROGER D. COVERLY, as
Florida registered voters,

and

GOVERNOR GEORGE W. BUSH and DICK
CHENEY, as candidates for President and Vice
President of the United States of America,

Plaintiffs,

vs.

THERESA LePORE, CHARLES E. BURTON,
CAROL ROBERTS, JANE CARROLL,
SUZANNE GUNZBERGER, ROBERT LEE,
DAVID LEAHY, LAWRENCE KING, JR.,
MIRIAM LEHR, MICHAEL McDERMOTT,
ANN McFALL, and PAT NORTHY, in their
official capacities as members of the County
Canvassing Boards of Palm Beach, Miami-Dade,
Broward and Volusia Counties, respectively,

Defendants.
______________________________________/

DECLARATION OF REBECCA T. MERCURI

Rebecca T. Mercuri, being first duly sworn upon oath and under penalty of perjury, does hereby state and affirm that if called to testify in the above captioned proceedings, she would testify as follows:

1. My name is Rebecca T. Mercuri. My title is President, Notable Software, Inc. of Lawrenceville, New Jersey, a computer consulting firm I founded in 1981. I am also a full-time member of the Computer Science faculty at Bryn Mawr College, Bryn Mawr, Pennsylvania. On October 27, 2000, I successfully defended my Ph.D. thesis at the School of Engineering and Applied Science at the University of Pennsylvania. The title of my thesis is: "Electronic Vote Tabulation Checks & Balances."

2. I hold a Bachelor of Science degree in Computer Science from The Pennsylvania State University, a Master of Science degree in Computer Science from Drexel University, and a Master of Science in Engineering degree from the University of Pennsylvania. My Ph.D. from the University of Pennsylvania will be awarded upon the final approval of the dissertation manuscript.

3. I have been involved with electronic vote tabulation since 1989, primarily in the capacity of an expert witness, but occasionally as a system examiner. Nearly all of this work has been pro-bono. My other expert witness work, which is for pay, has been through Notable Software, Inc. Some of my clients have included: The Public Defender's Office, State of New Jersey; The Office of Attorney Ethics, State of New Jersey; and numerous private law firms in Pennsylvania and New Jersey. Cases in which I have been involved include: criminal investigations, civil and municipal matters, product performance claims, and patent reviews. My expert witness work has, on occasion, involved forensic collection and examination of physical evidence (such as data media, computer hardware, and software), and review and reconstruction of damaged or deleted files.

4. I have published numerous papers and articles regarding voting machines over the last decade in a variety of forums, including the National Institute of Standards and Technology's National Information Systems Security Conference proceedings, and the Association of Computing Machinery's Communications. Many of my papers are available on my website at: www .seas.upenn.edu/~mercuri (follow the electronic voting link and also the [somewhat outdated] interactive resume link). My writings have focused on the flaws in computerized vote tabulation and ballot casting systems, primarily those related to auditability (a major factor in recounts) and anonymity (privacy).

5. I was heavily involved in expert testimony through the 1990's regarding the New York City voting system procurement. For that project, I read the extensive requirements for purchase as well as the subsequent system evaluation documents, and provided comments and sworn testimony at numerous New York City Board of Elections hearings on the ability (or inability) of the proposed systems to meet the stated requirements. (The $60M purchase was ultimately cancelled.) During 1993 I examined documents and provided comments pertaining to the mayoral election in St. Petersburg, Florida, where significant anomalies in the vote tally ultimately resulted in a manual recount. Over the years, I have also provided comment on voting system procurements in Pennsylvania, Nevada, and Hawaii.

6. Although I am new to the Bryn Mawr faculty as of this year, I have served as a Professor of Computer Science at other colleges and universities, including Drexel University in Philadelphia. My specialties in the computer science field are: computer

languages, computer architecture, computer-related risks, and digital multimedia. Through Notable Software I also led training sessions in computer languages and applications for such agencies as the Philadelphia Stock Exchange, the Federal Aviation Administration, and the U.S. Army (at Fort Dix).

7. Prior to my return to academics, I was employed as a computer scientist and computer engineer in industry, working for such companies (either as a regular employee or consultant) as: Intel Corporation, AT&T Bell Laboratories, Merck Corporation, Sarnoff Corporation, and RCA Laboratories. My responsibilities for those companies included computer software and hardware analysis, design, user interfaces (ergonomics), engineering, production, and product testing. I am fluent in many computer languages, ranging from assembly through object-oriented programming, and am conversant with various operating systems. I can read circuit diagrams both digital and analog, and am also an amateur radio licensed operator (KA3IAX General Class).

8. All voting systems (from punched-cards to internet balloting) are inherently flawed. These flaws include technological problems (such as failures in hardware, like card readers), sociological problems (from ballot-switching to subversive computer code in tabulation systems), and socio-technical problems (like legislation that fails to adequately address technologies that are deployed for use in elections). For example, it is my understanding that Florida Election Law requires that card readers be certified to process cards with an error rate of only one in a million reads. This law fails to address the ballot error rate that involves the punching of cards by the voters, a rate that is well known to exceed 2% (and sometimes as much as 5%) of the votes cast for typical

elections. These ballot errors include the well-known problem of "hanging chad," misaligned cards, and ballots with holes that are not completely punched.

9. The cards that are read through the readers for pre- and post-election purposes are (to my understanding) pristine sample cards (that also may include defects so the code that rejects double-punched ballots can be exercised), but not the typical set of poorly-punched ballot cards produced by voters, which is actually the norm on election day. The true cards would have the hanging chad problem, which results in re-reads producing different result totals.

10. Given the card-reader manufacturer's claim (which I have discerned through media reports) that after four re-reads, the ballots will stabilize such that hanging-chad will not be a problem, the election law should also specify that all cards be read through four times, in order to produce the most accurate count possible when using a machine. The ballot cards at issue here have only been sent through the reader at most twice, with resulting inaccuracies and undercounts due to hanging chad and other ballot errors. This is but one example of the many socio-technical problems to which I referred above.

11. All voting systems should have redundant mechanisms whereby each voter could verify the content of the ballot that they cast, thus providing the "checks and balances" that are critical to the democratic process. With punched cards, it has long been recommended (by Roy Saltman in his 1988 NIST treatise on Accuracy and Integrity in Computerized Vote-Tallying, and others, including myself) that paper balloting systems (punch-card and mark-sense) should contain identification of the candidate

directly on the ballot itself so that each voter can visually verify that their ballot is correct before placing it into the box.

12. Many States and municipalities have done this with their ballots, but apparently those being scrutinized in Florida have not, despite the fact that it was well known that this would effectively lower the voted ballot error rate, and also despite the enormous problems experienced in Florida with ballot problems during the 1988 Senate election. One might think that this is all water under the bridge, but since the voters were not given the opportunity to scrutinize their ballots, other humans should be permitted to do this. Indeed, election workers should be charged with the task of visually checking the ballots, in order to accurately ascertain the intent of voters where possible. The goal should be to determine the true will of the voters, not to slavishly cling to a machine count that is not fully accurate.

13. The cards themselves are the physical evidence of the election. They are, in fact, both the audit trail for the election as well as the expression of the intention of how each voter cast his or her ballot. In the truest sense, they represent the material upon which the forensic investigation of the election must take place. Since the card reader does not have the ability to do anything with the cards other than to verify that the ballot is "correct" (i.e. not double-punched in the same office) and to tally votes on those cards deemed correct, it is up to humans to inspect each individual card to determine what votes were actually cast.

14. All voting systems should provide an unimpeachable audit trail. In this case, it is the ballot cards themselves that provide this audit trail. The value of this fail-safe

mechanism is rendered meaningless if it is precluded from use when verifying election results. This would be akin to saying that the corpse could not be examined for the cause of death in a murder investigation.

15. I must strongly urge you, therefore, to permit the manual investigation of the paper ballots to proceed. Given the critical importance of the Presidential vote in Florida, it would be inappropriate to enjoin the requested and ongoing manual hand recounts, which will proved a more accurate reflection of the votes cast on Election Day. Having been an election worker at the polls for over 15 years, first in Pennsylvania and later in New Jersey, I know the seriousness with which each individual charged with the task of tallying the election takes their responsibility. It is my belief that the humans who will examine the cards will provide a better interpretation of the will of the voters than any machine can currently produce.

I declare under penalty of perjury that the foregoing is true and correct. Executed on November 13, 2000.

Rebecca T. Mercuri
Lawrenceville, New Jersey

Bibliography

[AND00a] Anderson, Mark K., *Close Vote? You Can Bid on It*, WiReD News, August 17, 2000.

[AND00b] Anderson, Mark K., *Voteauction Bids the Dust*, WiReD News, August 22, 2000.

[BAA88] Baase, Sara, *Computer Algorithms*, 2nd Ed., Addison-Wesley, 1988.

[BAE91] Baer, James, et. al, *Evaluation of offerors for the procurement of an electronic voting system*, Updated Evaluation of the Sequoia Pacific EVM (Environmental/Engineering Requirements), Prepared for New York City Elections Project, SRI International Project No. 7070, June 19, 1991.

[BAQ90] Baquet, Dean, with Gottlieb, Martin, *A Perennial Maze, New York's Election System*, New York Times, October 18 - 21, 1990.

[BEA99] Beardsley, Tim, *No Secrets*, Scientific American, Volume 281, Number 6, December 1999.

[BLA00] Blaze, Matt, and Bellovin, Steven M., *Tapping On My Network Door*, Communications of the ACM, Volume 43, Number 10, October 2000.

[BRA93] Brace, Kimball W., and Election Data Services, Inc. Staff, *The Election Data Book*, Bernan Press, 1993.

[BUR85] Burnham, David, *Vote by Computer: Some See Problems*, New York Times, August 21, 1985.

[CAI00] California Internet Voting Task Force, *A Report on the Feasibility of Internet Voting*, January, 2000. With Draft *Standards for Certification of Internet Voting Systems*, February 14, 2000. http:// www.ss.ca.gov/executive/ivote/home.htm

[CAL01] The Caltech/MIT Voting Project, *A Preliminary Assessment of the Reliability of Existing Voting Equipment*, February 1, 2001.

[CCI99] Common Criteria Implementation Board, *Common Criteria for Information Security Evaluation*, Parts 1, 2 and 3, Version 2.1, 1999. (Also known as ISO IS 15408.) http:// csrc.nist.gov/cc

[CLA00] Clark, David, *Encryption Advances to Meet Internet Challenges*, IEEE Computer, Volume 33, Number 8, August 2000.

[COH84] Cohen, Fred, *Computer Viruses - Theory and Experiments*, 1984. www.all.net/books/virus

[DUG88] Dugger, Ronnie, *Annals of Democracy (Voting by Computers)*, New Yorker, November 7, 1988.

[DUG92] Dugger, Ronnie, *The Virus That Ate Our Votes*, New York Newsday, August 26, 1992.

[DUG94] Dugger, Ronnie, *A Tale of Weird Drop-Offs and Jump-Ups: Are Computer Vote Counts Honest?*, The APF Reporter, Volume 16, No. 3, 1994.

[DUG00] Dugger, Ronnie, Democracy Under Stress, Los Angeles Times, November 19, 2000.

[FEC90] *Performance and Test Standards for Punchcard, Marksense and Direct Recording Electronic Voting Systems*, Federal Election Commission, 1990.

[FEC90a] *FEC Approves Voluntary Computer Voting System Standards*, Press Release, Federal Election Commission, Washington, DC, January 25, 1990.

[FRE88] Frenkel, Karen A., *Computers and Elections*, Communications of the ACM, Volume 31, Number 10, October 1988.

[FRE96] Freier, Alan O., Karlton, Philip, and Kocher, Paul C., *The SSL Protocol* (Version 3.0), Netscape Corp., March 1996. http:// home.netscape.com/eng/ssl3/ssl-toc.html

[GEL01] Gelles, Jeff, *N.J. critic says booth proved not so fail-safe*," Philadelphia Inquirer, January 15, 2001.

[GTECH] For the Engineering Doctoral Student, dissertation guidelines published at Georgia Tech, based on an earlier anonymous document distributed by the faculty at Northwestern University.

[ISF97] International Information Security Foundation, *Generally-Accepted System Security Principles*, Version 1.0, June 1997. http://web.mit.edu/security/www/GASSP/gassp021.html

[LED00] Ledbetter, James, *Net Out the Vote*, The Industry Standard, March 27, 2000.

[MAN00] Mannix, Margaret, *The Web's Dark Side*, U.S. News and World Report, August 28, 2000.

[MER91] Mercuri, R. T., Questions for Electronic Voting Machine Vendors, Urban Policy Research Institute, Election Watch, February 1991.

[MER92] Mercuri, Rebecca T., *Physical Verifiability of Computer Systems*, 5th International Computer Virus and Security Conference, March 1992.

[MER92a] Mercuri, Rebecca, *Voting-Machine Risks*, Inside Risks, Communications of the Association for Computing Machinery, Volume 35, No. 11, November 1992.

[MER93] Mercuri, Rebecca, *The Business of Elections*, 3rd Conference on Computers, Freedom and Privacy, March 1993.

[MER93a] Mercuri, Rebecca, *Threats to Suffrage Security*, 16th National Computer Security Conference, September 1993.

[MER93b] Mercuri, Rebecca, *Corrupted Polling*, Inside Risks, Communications of the Association for Computing Machinery, Volume 36, No. 11, November 1993.

[MER00] Mercuri, Rebecca, Declaration in App. of Appellee-Intervenor Florida Democratic Party in *Siegel*, No. 00-15981-C (CA11) tab 16, ¶ 9, dated November 13, 2000.

[MEY79] Meyers, Glenford J., The Art of Software Testing, John Wiley & Sons, 1979.

[MOO00] Mooney, Richard E., *A Vote for New Voting Machines*, Gotham Gazette, Citizens Union Foundation, July 17, 2000.

[NCS85] National Computer Security Center, *Department of Defense Trusted System Evaluation Criteria (TCSEC)*, DOD-5200.28-STD (Orange Book), December 1985.

[NCS90] National Computer Security Center, *Trusted Product Evaluations, A Guide for Vendors*, NCSC-TG-002, June 1990.

[NCS91] National Computer Security Center, *Integrity-Oriented Control Objectives: Proposed Revisions to TCSEC*, C Technical Report 111-91, Library No. S238,183, October 1991.

[NCS92] National Computer Security Center, *Trusted Product Evaluation Questionnaire*, NCSC-TG-019, May 1992.

[NEU89] Neumann, Peter G., and Parker, Donn B., *A Summary of Computer Misuse Techniques*, 12th National Computer Security Conference, October 1989.

[NEU94] Neumann, Peter, Reeve, Lance, and Webb, Douglas, *EVS Security and Control Report Review*, Prepared for New York City Elections Project, SRI International, July 14, 1994.

[NEU95] Neumann, Peter G., *Computer Related Risks*, Addison-Wesley, 1995.

[NEU00a] Neumann, Peter G., *Denial-of-Service Attacks*, Communications of ACM, Volume 42, Number 4, April 2000.

[NEU00b] Neumann, Peter G., *Robust Nonproprietary Software*, IEEE Symposium on Security and Privacy, May 2000.

[ONE87] O'Neill, Tip, with Novak, William, *Man of the House*, Random House, 1987.

[PER97] Perens, Bruce, *Open Source Definition*, June 1997.
http:// opensource.org

[PHI00] Phillips, Deborah M., and Jefferson, David, *Is Internet Voting Safe?*, Voting Integrity Project, July 10, 2000. http:// www.voting-integrity.org/text/2000/internetsafe.shtml

[SAL88] Saltman, Roy G., *Accuracy, Integrity, and Security in Computerized Vote-Tallying*, U.S. Department of Commerce, National Bureau of Standards, NBS (now NIST) Special Publication 500-158, August 1988.

[SAL93] Saltman, Roy G., *Assuring Accuracy, Integrity and Security in National Elections: The Role of the U.S. Congress*, Third Conference on Computers, Freedom and Privacy, CPSR, March 1993.

[SAM92] Sammon, Bill, *False election results probed*, The Plain Dealer, June 11, 1992.

[SCH00] Schneier, Bruce, *Secrets and Lies: Digital Security in a Networked World*, John Wiley & Sons, Inc., 2000.

[SHA93] Shamos, Michael Ian, *Electronic Voting -- Evaluating the Threat*, Third Conference on Computers, Freedom and Privacy, CPSR, March 1993.

[SHE98] Shea, Kevin, *Staggering county error alters result*, The Trenton Times, November 4, 1998.

[STE92] Stephens, Mary, *New voting machines don't satisfy county*, The Columbus Dispatch, June 12, 1992.

[THO84] Thompson, Ken, *Reflections on Trusting Trust*, Communications of the ACM, Volume 27, Number 8, August 1984.

[TRO89a] Trombley, William, *Paper Ballots' Days May Be Numbered*, Los Angeles Times, July 3, 1989.

[TRO89b] Trombley, William, *Electronic Elections Seen as an Invitation to Fraud*, Los Angeles Times, July 4, 1989.

[TRO89c] Trombley, William, *Computers: Bugs in the Ballot Box*, Los Angeles Times, July 2, 1989.

[US] *Constitution of the United States of America*, Amendments 14, 15, 19, 24, 26.

[US00] Supreme Court of the United States, *On Writ of Certiorari to the Florida Supreme Court*, No. 00-949, December 12, 2000.

[WEI00] Weinstein, Lauren, *Risks of Internet Voting*, Communications of the ACM, Volume 43, Number 6, June 2000.

[WIE93] Wiener, Lauren Ruth, *Digital Woes: Why We Should Not Depend on Software*, Addison-Wesley, 1993.

[WOL00] Wolf, Richard, *Arizona voters click into history*, USA Today, March 10-12, 2000

[WSJ00] The Wall Street Journal, *Election.com Aims to Revolutionize The Voting Process With Online Ballots*, May 8, 2000.

Index