ARTICLES

ACCURACY, INTEGRITY AND SECURITY IN COMPUTERIZED VOTE-TALLYING

The following excerpts have been gleaned from a 130-page report of potential inaccuracies and fraud in computerized voting systems. Recent difficulties in automated vote-tallying, including specific legal cases, are detailed along with a summary of conclusions and recommendations.

ROY G. SALTMAN

There are several examples of difficulties in the use of computerized vote-tallying, that occurred in the years 1980–1986. Two such cases took place in Carroll County, Maryland and Stark County, Ohio. It should be noted that the following descriptions are not intended to be the definitive versions of what occurred. No absolute proof can be offered that the events occurred exactly as described. However, reported data have been supplied by named sources on the scene, and exact quotations by participants and observers are given when appropriate.

Carroll County, Maryland: November, 1984

Carroll is a county of about 100,000 population located about 30 miles northwest of the city of Baltimore. On November 8, two days after the Tuesday, November 6, 1984 general election, and in accordance with the rules of the Maryland State Administrative Board of Election Laws (SABEL), voted punch card ballots from two districts of Carroll County were taken to a neighboring county, Frederick, to be rerun on an independently-managed system. (Similarly, ballots from Frederick County were taken to Carroll to be rerun.) This rerun is necessary under Maryland regulations to verify the original results before certification.

It was clear from these reruns that one of the computers used was in error in determining the outcome of a contest between Wayne Cogswell and incumbent T. Edward Lippy, for Carroll County School Board. Manual counts of the votes on ballots from both Frederick and Carroll counties showed that the Carroll County computer was the one that was incorrect. The initial but unofficial count, made public on the evening of the election, had indicated incorrectly that Cogswell was the winner.

An investigation undertaken the next day (November 9) by Craig Jester, a county computer program contractor, demonstrated that a wrong utility computer program for reading the ballot cards had been used. After the correct utility program was installed, the results coincided with those obtained manually and with the Frederick County computer. The utility program, named COLBIN, had been previously written by Jester under contract to the county and had been successfully used in the May, 1984, primary election.

The purpose of the COLBIN utility program was to read the voted ballot cards in the “column binary” format used for voting, rather than in a simpler format. At the request of Carroll County DP personnel to reduce the price, Computer Election Systems, the vendor of the vote-tallying system, had supplied the system with an elementary utility program that could read cards only in the simpler format. With this format, ballot
The two contests were listed on different ballot pages of were for Lippy, because many Lippy voters chose the position in the same column created a punch configuration recorded in both the school board and home rule issue the "votomatic" ballot holder. The combination of votes home rule position listed in the same card column. Votes not recorded on the home rule issue did not the same punch card columns as a home rule issue. Lots did not reveal the error.

The incorrect announcement was not due to any error in the vote-tallying computer system supplied by the primary vendor, Computer Election Systems, nor any activity undertaken by its representatives. Nevertheless, on July 29, 1985, the New York Times, in referring to this particular situation, reported that "The vote counting program that has been challenged in ... Maryland was developed by Computer Election Systems of Berkeley, Calif." [2].

The error was discovered after the election but before certification because of a Maryland regulation that required recounting on an independently managed system. This specific regulation was based on a recommendation that "further confidence in the machine-counted results can be achieved if mandatory machine recounting of a percentage of the precincts for each race is carried out on a different, independently managed computing system than that used to produce the official count" [8].

On June 11, 1985, another recount of the school board race in question was carried out, using the Carroll County computer, again including the COLBIN program. This recount was undertaken at the request of the State court in which Mr. Cogswell, the defeated candidate, had filed a suit asking that the results of the election be re-examined. The recount verified the correctness of the election results certified on Nov. 14, 1984, although Mr. Van de Bussche has indicated that the recount results did not exactly match the count reported in the certification.

Mr. Van de Bussche has stated that the recount, carried out with all sides in attendance, was hurried and less than precise in that, with the permission of the court, card reader "checks" were ignored in the ballot-reading process. Usually a card reader "check," indicative of a reading failure, would result in a decision to re-read the entire precinct of voted ballot cards. Instead, the card or cards causing the "check" remained unread and the reading process continued. The failure to re-read an entire precinct upon occurrence of a read "check" resulted in a small but random reduction of votes to both candidates, according to Mr. Van de Bussche. The differences were not significant enough to raise reasonable doubt as to the correctness of the certified results.

According to a July 11, 1985 story by Chris Guy in the Carroll County Times referring to the court-ordered recount, "... defeated candidate Wayne Cogswell had verification that use of an incorrect computer program caused a nearly 13,000-vote mistake in the unofficial totals released election night" [4].

Stark County, Ohio: May, 1986
The following description is adapted from the account given in the July 21, 1986 issue of Election Administration Reports [1], Richard G. Smolka, editor, with permission of the publisher.

Stark is a county of about 400,000 population located about 60 miles south-southeast of Cleveland. An un-
preceded court-ordered "audit" (hand recount) of a Stark County computer recount in a county commissioner's primary contest again named as winner the candidate who had apparently won in the official results of the May 6, 1986 primary but lost in the computer recount. The audit revealed a computer program error that permitted over 100 invalid punchcard ballots to be counted in the recount.

At the end of the election-night count, Robert A. Capestrain held a 26-vote lead in the three-person contest to be Democratic nominee for county commissioner. A recount by computer on May 27 (held because of the closeness of the original tally) put Patty Miller ahead by 5 votes. For the computer recount, the computer program used to obtain the original results was not used. Instead, a special computer program was written, in order to count only the disputed contest and not other contests on the ballot. The mystery, however, was why 165 additional votes had been tallied in the recount although the number of ballots read by the computer was the same.

The following table provides the votes for the three candidates in the computer tally of the primary, the computer recount, and the hand counted audit:

<table>
<thead>
<tr>
<th>Candidates</th>
<th>Primary May 8</th>
<th>Recount May 27</th>
<th>Change</th>
<th>Audit July 8</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robert A. Capestrain</td>
<td>12,967</td>
<td>13,018</td>
<td>+ 51</td>
<td>12,980</td>
<td>- 38</td>
</tr>
<tr>
<td>Donald E. Casar</td>
<td>7,987</td>
<td>8,019</td>
<td>+ 32</td>
<td>7,980</td>
<td>- 30</td>
</tr>
<tr>
<td>Patty Miller</td>
<td>12,941</td>
<td>13,023</td>
<td>+ 82</td>
<td>12,952</td>
<td>- 71</td>
</tr>
<tr>
<td>TOTALS</td>
<td>33,895</td>
<td>34,060</td>
<td>+165</td>
<td>33,921</td>
<td>-139</td>
</tr>
</tbody>
</table>

The 165 additional votes in the recount were randomly distributed throughout the 481 precincts. Most precincts had no changes, and most of those with changes had a one-vote increase. All candidates gained votes. The names of the candidates were rotated by precinct in the ballot booklets in positions 98-100-102, and the extra votes were distributed among these numbers on the ballot cards. Each of the three positions received approximately the same number of additional votes.

Initially, there seemed to be no satisfactory explanation for the additional votes. Hanging chad (bits of paper remaining after holes are punched) was suspected as a possible cause. Fraud was much less likely because it would have required access to ballots from all affected precincts, working knowledge of the ballot rotations, plus sufficient time to locate and punch ballots which had not been voted for county commissioner.

Following the computer recount that indicated a reversal of the initial count, candidate Capestrain filed suit challenging the recount. Because of the unusual nature of the recount result and rumors of fraud, the candidates, attorneys, election board, and court agreed to audit procedures that would resolve any identifiable problems with hanging chad as well as ensure that the vote count would be complete and accurate. Most importantly, they all agreed that the audit would constitute a final resolution of the vote count dispute. Judge Harold E. DeHoff of the Stark County Court of Common Pleas included a provision in the agreement that all parties would waive any rights of appeal.

The audit included a manual count and a computer count. A large number of 2-person teams were assigned to manually count the ballots under specific rules. The court order also provided guidelines on removal of hanging chad and specified that only the two master commissioners, appointed by the court, could remove a suspected chad or hanging chad.

Before the start of the audit, Ohio Director of Elections Dorothy Woldorf and area manager Robert Braun of the vote-counting system vendor gave the counting teams both written and verbal instructions on procedures to be followed. The manual recount began at 9:00 a.m. and continued until completion at about 7:45 p.m.

After the first several precincts were manually counted, it became evident that the audit was producing totals more closely matching the original count rather than the recount. By 11:00 a.m., the recount program error had been uncovered. The error was due to the failure of the recount program to distinguish between Democratic, Republican, and unaffiliated ballots.

In the May 8 primary, voters were given Democratic, Republican, or unaffiliated ballots, depending on their party registration. The logic in the computer program and associated header cards that were used to tally the primary ballots was able to distinguish among the different types of ballots, even though all the ballots were tallied on the same computer equipment.

In the recount, all the ballots were again tallied together on the same equipment, but the logic of the recount program could not distinguish among the different ballot types. It was apparently believed by the author of the recount program that the assignment of unique ballot positions to each contest and candidate was sufficient to separate the ballots. However, some Republican and unaffiliated voters had "voted" (i.e., punched out chad) in a ballot position assigned to a candidate in the Democratic county commissioner contest. These ballots were not counted in the Democratic primary tally, but they were counted by mistake in the computer recount.

In the audit on July 8, the ballots were first separated by party before being given to the two-person teams. The separation was easily accomplished because the
ballot types were distinguishable by color. Consequently, in the manual recount, “votes” by Republican and unaffiliated voters were not tallied.

During the audit, the master commissioners completed removal of chad on 28 ballot cards. Nine of these were identified as hanging chad, and the others were termed “bulging chad.” One commissioner said that it was obvious that the voter had detached the chad, but that it had been pressed back into position, probably when the cards were stacked. The removal of the chad by the commissioners had no effect on the outcome, but did increase the vote by a net of 26 over the original count.

“The computer program that was used to count more than one-third of the votes cast in the Presidential election last year is very vulnerable to manipulation and fraud, according to expert witnesses in court actions challenging local and Congressional elections in three states . . .”

—New York Times
July 29, 1985

SUMMARY OF CONCLUSIONS AND RECOMMENDATIONS
This report has been prepared with funding provided by the John and Mary R. Markle Foundation of New York City. The Markle Foundation requested that the National Bureau of Standards (NBS) undertake this study because of public concern about the potential for inaccuracy or fraud in computerized vote-tallying. NBS was approached because of its experience with the subject matter as the result of a previous project undertaken by the author for the U.S. General Accounting Office.

Concern had been heightened by a series of articles published in the summer of 1985 in the New York Times. The articles cited statements by two computer experts reporting that a computer program widely used for vote-tallying was vulnerable to tampering. Several elections were identified in which losing candidates claimed that it would be possible to fraudulently alter the computer programs that were used in their contests.

In preparation for this report, a review of recent public statements and documents was undertaken that indicated concern about computerized vote-tallying. The review showed that the problems could be categorized as follows: there is difficulty in verifying results; there is the possibility of undiscoverable frauds; and election administrators lack the necessary knowledge and resources.

There is a continuing problem of public confidence. While proof of actual computer program manipulation appears to be lacking, documentation conclusively demonstrating otherwise is generally insufficient, due to the manner in which many computerized elections are conducted. It has been clearly shown that audit trails that document election results, as well as general practices to assure accuracy, integrity, and security, can be considerably improved.

Government Responsibilities
The recommendations that respond to these problems are directed to state and local government election officials. Elections for state and federal offices are conducted by local government (generally county, township, and city) administrators. In about one-third of all counties, including over one-half of all registered voters, voting is carried out using computerized equipment. The local administrators require the necessary resources and expertise to efficiently and effectively carry out their responsibilities.

These responsibilities generally include procurement of vote-tallying systems and supporting services. An effective procurement must include the development of specifications so that accuracy, integrity, and security will be assured. The local administrators also have the responsibility for implementing the necessary management control systems to enable the public to have confidence in the results produced.

Election officials require a source of neutral expertise for the receipt of new technical and administrative information. The establishment of the Election Center in the Academy for State and Local Government clearly fulfills a need. Its efforts should be expanded.

Implementation of an Internal Control Function
Essential recommendations are that the concept of internal control should be extended so as to be applicable to vote-tallying, and that persons knowledgeable in that professional field should be utilized to assist in the establishment and implementation of sound operational procedures. To the extent that computerized voting equipment and software must have supporting capabilities, these procedures should be reflected in procurement specifications.

Internal control, which is nearly universally used as a management technique in financially oriented applications (e.g., in banking or manufacturing, or to safeguard assets in any type of organization), has not been applied to vote-tallying because those operations are not priced. Applicability of internal control to vote-tallying requires only the redefinition of the concept of a transaction. A transaction is now defined as a business event that is measured in money and that is en-
Articles

In order to assure accuracy, integrity, and security in vote-tallying, best left to individuals who are professionally qualified and personally knowledgeable about specific technical matters. With the addition of needed technical resources to the staff, election administrators would be able to retain management control, and not have to abdicate it to others, such as vendors or data processing center directors. Thus, election administrators would be able to retain the capability of directing that the objective of assuring accuracy, integrity, and security in vote-tallying be carried out.

FEC Clearinghouse Specifications

The performance specifications developed by the National Clearinghouse on Election Administration of the Federal Election Commission (FEC Clearinghouse) are approaching completion, and they are intended for statewide adoption. Each state should consider the adoption of these specifications when they are issued. Acceptance procedures for hardware and software should be consistent with the FEC Clearinghouse implementation plan for adoption of these specifications. That plan calls for qualification and certification prior to final acceptance. Qualification implies conformance with standards and functional requirements, and may be done once to satisfy many states. Certification ensures that the product meets state requirements. Acceptance testing evaluates the degree to which the specific units delivered to the local government conform to approved characteristics.

Revised Texas Statute on Electronic Voting Systems

The requirements of the revised Texas statute on electronic voting systems should be considered for adoption in those states that have not already adopted equivalent or more stringent provisions. Requirements of the Texas statute include audit trails, deposit of computer

"Extra votes may be entered in the form of bogus ballots on punch cards, or vote totals may be altered through the use of control cards," said (Howard) Strauss (associate director of the Princeton Computer Center). "Either of these assaults on the system could be performed successfully by a computer novice."

—New York Times
July 29, 1985
and acceptable dimensions can be created by a voter using the new stylus without the need for pre-scoring. The internal construction of the "votomatic" ballot holder must be altered with the use of the new stylus. Other devices and methods for elimination of pre-scored punch card ballots also may be effective.

If a ballot cannot be read by machine, administrative controls should be in place to permit such ballots to be counted manually. A voter's choices should not be lost because of machine failure.

Testing to determine the accuracy of current ballot reading systems (such as that now being carried out by ECRI of Plymouth Meeting, PA), and research to improve ballot tallying systems in accuracy and ease of voter use, are important to pursue.

Design of Direct Recording Electronic (DRE) Machines

With DRE machines, no ballot is used. The voter enters choices directly into a storage unit of the machine with the use of pushbuttons, a touch screen, or similar devices. As no voter-generated records of choices exist, and no recount independent of the machine is possible, steps should be taken in the design of these machines to assure complete confidence in the reported results.

A problem with most DRE machines as currently designed (as with lever machines, their predecessors), is that there is no difference in the results seen between a voter's failure to cast a vote and the machine's failure to record a vote.

Recording of Undervotes. It is recommended that each DRE machine be designed so as to take a positive action indicating a "no vote" for every choice that the voter fails to take. When voting is complete, the voter's choices, and any "no votes" for votes not taken, would be transferred to a more permanent storage for summation with other voters' choices. The required transfer and summation of the "no votes" would serve as positive indications of the voter's failure to make certain specific choices. Thus, there would be no ambiguity about whether the voter failed to vote or the machine failed to record selections.

Retention of Voter-Choice Sets for Summation Verification. Each voter-choice set (i.e., the machine's record of all choices of a voter) should be retained in the machine on a removable non-volatile medium (e.g., magnetic disk). Storage locations of the voter choice sets would have to be randomized to prevent association of a particular set with a particular voter. The retention of the voter-choice sets makes possible a verification (on an independent machine) of the DRE machine's summation of the voters' choices that it recorded. The correctness of the machine's data entry process cannot be checked in this manner.

DRE data entry hardware should be certified for logical correctness, by examination of the logic design and by testing under a large variety of different conditions. The DRE data entry function must be correct, as there are no ballots to provide an independent check. The data entry logic and its documentation should be deposited with the state.

(Software recommendations are discussed in the accompanying box "Vulnerabilities of Vote-Tallying Software" beginning on p. 1190.)

RECOMMENDATIONS ON OPERATIONAL PROCEDURES

Lack of sufficient pre-election testing appears to be a major source of operational difficulty. Sufficient pre-election testing should be done so that errors in software specialization or in implementation of logical rules, if any, will become obvious. It is recommended that to the greatest extent possible, all hardware and software to be utilized should be given a dry run simulating specific conditions to be faced on election day and election night.

Audit Trails. Audit trails provide the supporting documentation through which the correctness of the reported results may be verified. Two types of audit trails are necessary to document operations and provide confidence in the results reported. One type records steps in the operation of the equipment, while the other records steps in the voting and vote-tallying processes.

Complete Data from Split Precincts. Each split of a split precinct should be treated like a separate precinct for the reporting of ballots and votes cast.

Access Controls. Access (i.e., security) controls must be in place during preparation for voting, voting itself, and vote-tallying. These controls concern access to sites, areas, facilities, equipment, documents, files, and data. The controls cover transportation of ballots and telecommunication of results.

Internal Controls for Tallying Systems. These controls should be in place to prevent all types of ballot frauds and miscounting errors, and to provide the documentation and assurance that the correct results are reported. The controls on ballots cover printing and distribution, accounting for use, validity, and prevention of errors.

"Many local election officials are baffled by computers and are unable to understand, question and challenge the computer systems."

—New York Times
August 21, 1985
VULNERABILITIES OF VOTE-TALLYING SOFTWARE

Software that is intended for use in more than one election is usually written in a generalized form, and then specialized for a specific election. The generalized software may be provided with a set of tables whose values are to be filled in during specialization.

This two-step process may be applied to any form of computerized system: punch card, mark-sense, or DRE (Direct Recording Electronic) machine. The specialization process establishes the number and names of offices and issues to be considered by the voters: the number of candidates and their names for each office; the number of allowable votes for multiple selection situations; and the implementation of special voting rules, such as for treatment of straight-party overvotes and for crossover voting. When the implementation for which the software is being provided includes precinct-located machines, the specialization process includes provision of the necessary parameters to complete the tables in the software of those machines. A device may be provided that can be used to copy the parameters of the election for each precinct from the main computer into the EPROM of the appropriate precinct-located machines.

Software vulnerabilities include logical errors, the possibility of hidden code, and undocumented changes. No system type is immune from these threats. Protecting the software requires implementation of effective management procedures that assure system integrity and security.

Logical errors. There are several categories of logical errors to be considered. First, a distinction may be made between generic logical errors that are independent of the conditions of elections, and election-related errors. Then, the latter may be subdivided into rule-implementation errors and setup-condition errors. Rule-implementation errors concern pre-established voting rules, such as for crossover voting and for straight-party overvotes. Setup-condition errors involve specific election matters, such as wrong rotations and incorrect assignments of candidates to districts.

Generic logical errors should be identified and eliminated during acceptance testing, when the software is first procured for use. (Acceptance at the state level, involving approval for subsequent use within the state, may constitute a certification.) Ideally, certification/acceptance activity should include a test of all possible features and functions and combinations. Resource constraints may limit testing to only some of the possible combinations of program features for a system. Certification/acceptance testing may involve review of logical flow charts and specific program code, as well as determination of system response to the normal flow of ballot data. Additionally, responses may be evaluated against the input of incorrect data, special cases, large volumes of data, and unplanned hardware failures while processing. Guidance in planning computer software acceptance testing is available [5], [10].

Certification/acceptance testing requires evaluation of the software as it will be implemented, so that elimination of rule-implementation errors should be a byproduct of this activity. A particularly difficult type of rule to implement is the required restriction in some states to one vote per candidate per voter in a vote-for-two situation when a candidate is listed on two party lines for the same office. Should a voter attempt to vote for the same candidate on both party lines in this situation, an incorrectly implemented DRE, punch card, or mark-sense system may assign the candidate two votes.

Elimination of setup-condition errors is necessary as an activity in pre-election system checkout. In precinct-count systems, it is important to verify that the computer is implemented for the same rotation and other setup conditions as seen on the ballot for that precinct. In central-count systems, testing the system against a large number of ballots or ballot images may bring to light the presence of setup-condition errors. In any system in which there is ballot rotation, it is important to assure that the summarization of each candidate's votes is correct, considering the different ballot position of the candidate in the various precincts.

Hidden code. This term refers to a secret computer program inserted into another program provided to an unaware user. The purpose of the hidden code may be malicious, or it may be used to demonstrate the capability of the perpetrator, or it may be used to surreptitiously record information for later retrieval. The effort of finding any hidden code within a vote-tallying program may be assisted with the use of a software engineering tool that identifies the use of each path in the program during program execution. If, in the exercise of a vote-tallying program, a particular path is not used (because the conditions that select it have not been employed), such a path may be further investigated to determine the selection conditions.

The problem of finding hidden code is complicated when vote-tallying software is mounted on a general-purpose computer. Hidden code may have been initially placed in a support program of the computer, for example the compiler or operating system. Such hidden code may be activated when the support program is called from the vote-tallying program. If a clock could be accessed by the hidden code, activation could be arranged to occur at a particular time on a particular day. The identification of the presence of such hidden code could be difficult and time-consuming, and a deliberate search, without specific evidence that the code exists, could be impractical.

Assuring System Integrity. Realistic protection against the presence of hidden code in a general-purpose computer installation involves the application of general administrative controls over the installation. An important administrative control for protection against hidden code is the requirement that software be supplied only from an original manufacturer who is known, reliable, and can be held accountable. The copying of software from secondary sources should be forbidden.

Modifications to software must be strictly controlled to prevent unauthorized changes. Restrictions must be in place to control access to program codes that have been approved for operation. Authorized changes should be documented. Previous versions of revised software should be retained according to established procedures. Due to the threats from hidden code, computer programs have been written that protect against it. The protective program is intended to run with an application, and to identify any modifications that are made. However, it is necessary that the application be initially free of any hidden code because only subsequent changes can be identified. In addition, the protective program could not prevent the unauthorized modification of itself that might be undertaken from outside the program, for example from the operating system.

An additional strategy to protect system integrity is to have the vote-tallying software provided by its developer.
with all necessary support software, including the operating system, as a complete package. No other software would be necessary. In this way, the complete software for vote-tallying is separated from the influences of all other software. It is common practice for software for precinct-located machines to be provided in this condition. In the future, states may require the complete package of software (for any type of vote-tallying system) to be deposited with the chief elections official or an escrow agent, so that the provision of a complete package of software is responsive to this anticipated request.

The provision of a complete software package enables an additional protection to be applied: the data authentication code (DAC) [3]. This code value, a binary number, is computed by first randomly selecting a certain key value, and then applying that key and a specific mathematical function to the software package. The key value is kept secret. The computation of the DAC (which contains the same number of bits as the key) protects the software against both accidental and intentional unauthorized data modification. To apply the function, the software package is treated as if it were a digitally-encoded message, as the DAC was developed to authenticate messages transmitted over communication lines. In application, the DAC would be generated for the original software deposited with the state, and would be generated also for the supposedly identical software used in an election. The two values of the DAC will be identical if the software packages are identical. (The software to be used in an election would have to be compared with the master copy before subsequent specialization for the election.)

Internal Controls for DRE Systems. These controls should be in place to provide documentation and assurance that the correct results are reported when DRE systems are used. The controls cover matching machine use with voter totals, vote reconciliations on each machine, recounting of voter-choice sets, and post-election checkout of machines.

FUTURE SYSTEMS

Although none of the computer difficulty situations (reviewed in this report) has provided solid evidence of computer program manipulation, the reviews have revealed the need for improvements in hardware and software performance and in operational procedures, and they have provided support for the need for institutional changes. Thus, the reviews have influenced the recommendations provided in this report.

Specific recommendations directly resulting from the reviews of difficulties include the recommendations on improved accuracy in ballot tabulation, elimination of pre-scored punch card ballots, assurance of the counting of ballots rejected by readers, provision of complete data from split precincts, and more thoroughness in pre-election checkout.

While vote-tallying using telephones or stations similar to automatic teller machines is technologically feasible, the decision to implement such a system must be based on more fundamental factors. Any installed system must meet political and economic requirements, as well as technical requirements of accuracy and reliability. Political needs include equal access by individuals,
the ability to verify registration, and the ability of the 
voters to vote in secret without intimidation. Internal 
controls must be implementable to demonstrate the 
correctness of the reported results. Benefits, such as 
increased voter convenience and possible improved 
participation rates must be compared against the costs 
of implementation.

REFERENCES
1. Audit of Recount Detects Error, Restores Stark County, Ohio Com-
misioner Victor. Election Administration Reports 16, 5 (July 21, 1986), 
2-4.
2. Dunham, D. "Computerized systems for voting seen as vulnerable 
3. Computer Data Authentication. Federal Information Processing Stan-
4. Guy, C. "Election system under fire nationwide." Carroll County 
Times (July 11, 1985), 1.
6. Kelly, S. "Human error reversed school board election results." Car-
roll Sun (November 18, 1984). 2.
7. Lewis, T.W., Exner, T.E., and Gossell, J.C. "Certain factors which 
involved the Carroll County Board of Election Supervisors, Either 
Directly or Indirectly, in Producing the Numerical Results of the 
General Election in Carroll County. November 6th. 1984." (Decem-
ber 4, 1984).
National Bureau of Standards, NBSIR 75-687, March. 1985. 13. "Re-
counting." 91.
9. Van de Bassche, T.J. Letter to Dr. Thomas Lewis, November 26, 
1984.